

Masteroppgave i geofag

Test og utvikling av flyt- algoritmer for steinskred

Morten Berg



UNIVERSITETET I OSLO

DET MATEMATISK-NATURVITENSKAPELIGE FAKULTET

Test og utvikling av flytalgoritmer for steinskred

Morten Berg



Masteroppgave i geofag

Studieretning: Naturgeografi, hydrologi og geomatikk

Institutt for geofag

Matematisk-naturvitenskaplig fakultet

UNIVERSITETET I OSLO

14. mars 2008

© **Morten Berg, 2008**

Veiledere: Bård Romstad (UiO) og Bernd Etzelmüller (UiO)

Dette eksamensarbeidet er publisert elektronisk i DUO – Digitale Utgivelser ved UiO

<http://www.duo.uio.no>

Det er også katalogisert i BIBSYS (<http://www.bibsys.no/>)

All rights reserved. No part of this publication may be reproduced or transmitted, in any form or by any means, without permission.

Forord

Oppgaven er en del av mastergraden i naturgeografi, hydrologi og geomatikk, ved institutt for Geofag, Universitetet i Oslo. Oppgaven ble gitt av Bård Romstad, som har vært veileder. Bernd Etzelmüller har også fungert som veileder.

Jeg vil takke mine veiledere for faglig rådgivning og motivasjon. Spesielt vil jeg takke Bård Romstad som har latt meg bruke mange av sine arbeider i denne oppgaven. En takk rettes til Ulrik Domaas ved NGI, for hjelp, og til Jan Ketil Røed ved institutt for Geografi ved NTNU, for å tatt i mot meg under mitt opphold i Trondheim. Takk til Jon Endre Hausberg for ASTER-terrengmodellen. Takk rettes også til Marit Carlsen ved institutt for Geofag. Din støtte og positivitet var uvurderlig i motbakkene.

En spesiell takk til Morten Ramberg for faglige innspill, uvurderlig inspirasjon og gode samtaler over kaffekoppen. Tusen takk også til Inger Berg, Karl Rasmus Jahnsen Berg, Jenny Jahnsen Berg, Hanne Jahnsen og Kjell Stensland for moralsk oppbakning og deling av husrom under arbeidet. En helt spesiell takk for den uvurderlige innsatsen til Pernille Berg på alle plan under dette arbeidet, bortsett fra det rent faglige. Uten din hjelp og støtte hadde denne oppgaven aldri sett dagens lys. Og til slutt en takk til Leonora Berg, som kom til verden under arbeidet med denne oppgaven, for å være et lysglimt når jeg trengte det mest.

Sammendrag

Oppgaven undersøker bruk av flytalgoritmer for skredforskning, med spesielt fokus på steinskred. Flytalgoritmer er i utgangspunkt et velbeskrevet fenomen innenfor hydrologien, så inspirasjon hentes fra dette fagfeltet. En egen algoritme, D3, blir beskrevet. D3 er en modifisering av D8-algoritmen, og tar høyde for at steinskred har et bevegelsesmoment, dvs. en motstand mot å forandre bevegelsesretning. Algoritmen valideres opp mot digitalisert skred og det blir funnet at den gir anvendbare resultater, men har begrensninger i forhold til D8. En videreutvikling av algoritmen basert på disse resultatene beskrives. En flytalgoritme hvor flytretningen er basert på aspektet, blir også beskrevet. Denne sammenlignes med D3, og det viser seg at den fungerer langt bedre på teoretiske flater, men at det er lite å vinne når den brukes på realistiske terrengmodeller. Det indikeres at denne algoritmen fungerer bedre enn D3 på terrengmodeller av dårligere kvalitet, men videre studier trengs. Det argumenteres ut ifra resultatene at det er lite å vinne på å innføre mer komplekse algoritmer for skredsimulering. Tilfeller hvor valget av videre flyt er uavgjort, fordi to eller flere resultater er like, belyses. Det blir funnet at dette fenomenet er av signifikant betydning, men at det oftest medfører liten konsekvens. I undersøkelsen av dette fenomenet implementeres D3R. Det viser seg at denne algoritmen kan brukes til å simulere spredning.

Innholdsfortegnelse

Forord	5
Sammendrag	6
Innholdsfortegnelse	7
Figurliste	9
1 Innledning	11
1.1 Bakgrunn	11
1.2 Problemstilling og hypoteser	14
1.3 Disposisjon av oppgaven	15
1.4 Begreper	16
2 Teori	18
2.1 Steinskred	18
2.2 Regional analyse	19
2.3 Skredmodellering	20
2.3.1 Empiriske modeller	20
2.3.2 Prosessbaserte algoritmer	23
2.3.3 Topografiske modeller	24
2.4 Terrengmodelltyper	24
2.4.1 Raster	24
2.4.2 TIN	26
2.4.3 Kontur	27
2.5 Flytalgoritmer	27
2.5.1 D8	28
2.5.2 Grid-effekt i D8	30
2.5.3 D16	31
2.5.4 MSF	32
2.5.5 Rho8	32
2.5.6 FD8	33
2.5.7 DEMON	34
2.5.8 Dinf	36
3 Metoder	37
3.1 Metodedesign	37
3.2 Grunnlagsdata	37
3.2.1 Terrengmodell fra Statens Kartverk	37
3.2.2 ASTER-terrengmodell	38
3.2.3 Skreddatabasen	38
3.3 Implementasjon av D3-algoritmen	38
3.3.1 Det bevegelige vinduet	40
3.3.2 Simulering av skredbanen og avbruddskriterium	46
3.4 Håndtering av ties	48
3.4.1 Implementering av D3R	48
3.4.2 Analyse av forekomst av ties	51
3.5 Implementasjon av D360-algoritmen	51
3.5.1 Framgangsmåte for sammenligning	53
4 Resultater	55
4.1 Sammenligning med virkelige skred	55
4.1.1 Skred som går oppover i terrenget	58
4.1.2 Begrensning ved lik høyde	59
4.1.3 Forskjell på D3 og D8	60

4.1.4	Grid-effekt	61
4.1.5	Avbrudd	61
4.2	Ties	62
4.2.1	Terskelverdier	64
4.2.2	Konsekvensen av ties	67
4.2.3	Konsekvensen av ties i virkelige skred	68
4.2.4	Simulering av spredning	70
4.3	Sammenligning av flytalgoritmer	71
4.3.1	Kunstige overflater	71
4.3.2	Sammenligning på en ASTER-basert terrengmodell	78
4.4	Hva er en modell?	80
5	Konklusjon	83
5.1	Videre arbeid	85
6	Kilder	86
Appendiks	89
	D3R - Slidetree	89
	D3: Slide	90
	D360: SlidePath360	93
	D360: getaspect	94
	Matlabfunksjonen flowdirection	95

Figurliste

Figur 1: Flytskjema for testing av hypoteser i oppgaven	15
Figur 2: Klassifikasjon av hurtig massebevegelse (Hestnes 1979)	19
Figur 3: Forholdet mellom høyde og lengde (Scheidegger 1973)	21
Figur 4: Sammenheng mellom h/l og volum (Scheidegger 1973)	22
Figur 5: Bevegelig vindu (Berry 2008)	25
Figur 6: Eksempel på en pit som utgjøres av en enkeltcelle	26
Figur 7: Terrengmodeller (a) kontur, (b) raster, (c) TIN (Etzelmüller og Sulebak 2000)	27
Figur 8: Helningsvinkel α	29
Figur 9: Vekting i diagonale retninger	29
Figur 10: Sammenligning av D8 og D16 (Dorren 2003)	31
Figur 11: FD8 allokere flyt til alle lavereliggende celler	34
Figur 12: Flyt i DEMON	35
Figur 13: "Flow-tube" i DEMON	35
Figur 14: Flytretningen på triangulære fasettplan (Tarboton 1997)	36
Figur 15: Oversikt over programmet	39
Figur 16: Implementering av retning i D3	40
Figur 17: Eksempel på retningsverdier for en gitt celleblokk	41
Figur 18: Retningsverdier i D3	42
Figur 19: Flytdiagram for valg av retning i D3	43
Figur 20: Begrensning i algoritmen	46
Figur 21: Rutinen Slide i D3 som håndterer individuelle skred	47
Figur 22: Rutinen slidetree håndterer tie-punkter	49
Figur 23: Lokalt koordinatsystem	50
Figur 24: tie-punkter (1 og 4)	50
Figur 25: Struktur for D360-algoritmen	51
Figur 26: Flytdiagram for D360-algoritmen	52
Figur 27: Oppdateringsverdiene i x og y-retningen.	53
Figur 28: Utvalgte skred i Sør-Norge	56
Figur 29: Utvalgte skred i Nord-Norge	56
Figur 30: Byrkjelostøylen (D3)	57
Figur 31: Langhammaren (D3)	57
Figur 32: Tjelle (D3)	57
Figur 33: Urdbøuri (D3)	57
Figur 34: Hysket (D3)	57
Figur 35: Skjærsura (D3)	57
Figur 36: Gumpedalen (D3)	57
Figur 37: Kjusakdalen (D3)	57
Figur 38: Digitaliserte skred i Geirangerfjorden (grønn) og D3 (gul).	58
Figur 39: Digitalisert skred (grønn), D3 (gul)	58
Figur 40: Simulering fra kanten av løsneområdet	58
Figur 41: D3 vises som gul linje og D8 som blå. I firkanten vises høydeverdiene til nabocellene	60
Figur 42: Helningsgrad (Slope)	61
Figur 43: Helningsretning (Aspect)	61
Figur 44: h/l og skredvolum i Norge (Romstad et al. 2006)	62
Figur 45: Konvergerende oppførsel	63
Figur 46: Divergerende oppførsel	63
Figur 47: Dramatisk følge av divergerende skredbane	64

Figur 48: Geirangerfjorden.....	65
Figur 49: Terskel = 0.....	65
Figur 50: Terskel = 0,02.....	65
Figur 51: Terskel = 0,05.....	65
Figur 52: Jostedalsbreen.....	66
Figur 53: SKT.....	66
Figur 54: ASTER	66
Figur 55: Konsekvens av ties	67
Figur 56: Konsekvens av ties	68
Figur 57: Byrkjelostøylen (D3R)	69
Figur 58: Langhammaren (D3R).....	69
Figur 59: Tjelle (D3R).....	69
Figur 60: Urdbøuri (D3R)	69
Figur 61: Hysket (D3R).....	69
Figur 62: Skjærsura (D3R).....	69
Figur 63: Gumpedalen (D3R)	69
Figur 64: Kjusakdalen (D3R).....	69
Figur 65: Tilnærming til MF	70
Figur 66: Spredning på divergerende overflate.....	70
Figur 67: Helning NW.....	72
Figur 68: Helning NNW.....	72
Figur 69: Bolle (D3).....	72
Figur 70: Bolle (D30).....	72
Figur 71: Kule (D3).....	72
Figur 72: Kule (D30).....	72
Figur 73: D3- Geiranger.....	73
Figur 74: D360 - Geiranger.....	73
Figur 75: D3 (gul), D30 (rød) – Geiranger.....	74
Figur 76: Byrkjelostøylen (D360).....	75
Figur 77: Tjelle (D360)	75
Figur 78: Kjusakdalen (D360).....	75
Figur 79: Skjærsura (D360).....	75
Figur 80: Gumpedalen A (D360)	75
Figur 81: Gumpedalen B (D360)	75
Figur 82: Sal-formet terrengmodell.....	76
Figur 83: Forskjellig tillat avvik.....	77
Figur 84: Innvikfjorden (D3). Gul (SKT), rød (ASTER).....	79
Figur 85: Innvikfjorden (D30). Gul (SKT), rød (ASTER).....	79
Figur 86: Hierarki på retningene som sjekkes.....	83

1 Innledning

1.1 Bakgrunn

De tre største steinskredene i Norge (Loen 1905, 1936 og Tafjord 1934) har krevd flere liv enn alle andre skred til sammen (Hestnes 1985a). Disse skredene hadde til felles at de hadde utløp i en fjord og genererte kjempebølger som mangedoblet de destruktive kreftene. Norge har en topografi og mange fjordarmer som tilsier at skadepotensialet er stort. I tillegg til den direkte faren fra skredet, kommer den indirekte faren som skredutløste tsunamier representerer (Harbitz 2002, Harbitz et al. 2003). Skred som treffer vannmagasiner kan også utløse kjempebølger som kan føre til overskylling (eng. *overtopping*) eller destruksjon av demninger (Bakkehøi et al. 2004).

I andre deler av verden som er preget av fjellandskap ser man lignende trusler, og ofte er steinskred bare et av mange faremomenter, for eksempel i Sveits, hvor isskred fra breer, debris flow fra oppdemte sjøer og skred som går ned i elveløp representerer en økende fare for befolkede områder (Huggel et al. 2004). Også i Norge er skred fra isbreer et potensielt problem (Hausberg 2008). Skredfare er etter alt å dømme et problem som vil øke i omfang, både i forbindelse med utbygging av infrastruktur og global oppvarming (Huggel et al. 2003b).

Tradisjonelt har skredforskning foregått innenfor rammene av geoteknikken. Studier har vært gjort på enkeltsteder, hvor man har kartlagt alle parametere, som bergrunn, helning, sedimentlag osv. Men forsøket på å generalisere resultatene og dermed også bruke de aktivt til å finne farlige områder på regionalt nivå har vært begrenset, og strander i stor grad på lite datautvalg og kompliserte skredbeskrivelser (Hestnes 1979, Dorren 2003, Romstad et al. 2006). For eksempel ble skredfaren ved Åkerneset i Sunnysfjorden i Norge oppdaget ved en tilfeldighet, hvor personer med lokalkjennskap var svært våkne på sprekkutvidelser og satte seg i forbindelse med de riktige innstansene (T. Eiken, pers. kom., 2005). Mange slike utsatte steder er kjent og kan underkastes detaljstudium hvor alle parameter blir kartlagt (Harbitz 2002, Blikra 2003, Bakkehøi et al. 2004). Men de dramatiske konsekvensene slike katastrofer vil ha, tilsier at man trenger å identifisere potensielle fareområder på forhånd. Fjellandskapet og antall fjorder og innsjøer i Norge er for stort til at man kan utføre åstedsbefaringer overalt. Derfor trenger man verktøy som kan identifisere utsatte områder.

GIS gir en mulighet til analyse av geografisk informasjon som tidligere tiders geografer ikke hadde. Med en stadig utvikling mot mer datakraft og lagringskapasitet, mer komplekse metoder og større datatilgang, representerer GIS de verktøyene som trengs for å analysere farepotensial i regionale kontekster (Dorren 2003). I særdeleshet er data fra satellittbilder en viktig kilde til informasjon (Metternicht et al. 2005). Disse verktøyene må ordnes inn i et metodeverk før de kan gi svar på spørsmålet om hvor de potensielle fareområdene ligger. Resultatet av et slik metodeverk er en førstehånds analyse, som kan identifisere de mest utsatte områdene (Huggel et al. 2003b, Jaboyedoff og Labiouse 2003, Huggel et al. 2004, Salzmänn et al. 2004, Romstad et al. 2006). Dermed kan man plukke ut områder for detaljanalyse og feltarbeid. Med andre ord er et slikt metodeverk bare ment å være en bit av puslespillet. Et slik metodeverk kan også bidra til å utvikle skredfarekart (Hestnes 1985b, Romstad et al. 2006).

I et metodeverk vil det inngå en terrengmodell som datagrunnlag. Områder som Norge og Sveits har veldefinerte kartdatabaser som er grunnlag for terrengmodeller. Men mange fjærntliggende områder, ofte i politisk betente områder, har liten eller ingen datatilgengelighet. Dermed vil fjærnmåling gi mulighet for regional skredanalyse også i slike områder (Kääb et al. 2003). Videre må metodeverket inneholde verktøy for å identifisere løseområder. Her kan både satellittdata og topografiske parametere beregnet ut fra terrengmodellen gi muligheten til å identifisere disse (Romstad 2001). Det neste steget består i å beregne utløpet av skredet, både i retning og lengde. Et vesentlig spørsmål vil være om skredet treffer en vannkropp. Da kan man forvente en tsunamilignende kjempebølge. Det finnes bølgemodeller som kan beregne forplantningen og effekten av slike kjempebølger, og de er blant annet avhengig av skredets treffpunkt og treffvinkel (Bakkehoi et al. 2004). Tilslutt kan man vurdere konsekvensen ved å inkorporere data om infrastruktur i modellen, for å få en konsekvensanalyse.

Fokuset i denne oppgaven er på det andre steget, beregningen av skredløpet. Dette kan igjen deles opp i to punkter, skredets bane og skredets utløpsdistanse. Steinskrud er en gravitasjonspåvirket massebevegelse nedover en fjærllside, som består av mange enkelelementer. Et annet skredfenomen, debris flow, har et svært høyt vanninnhold og har mange likheter med en vannflom. Et løsmasseskrud eller et steinskrud kan i så måte sees på som en svært tyktflytende væske. Derfor er det nærliggende å hente inspirasjon fra et annet

fagfelt, nemlig hydrologien. Innenfor hydrologien er man opptatt av å bestemme flyt, fordi det inngår i andre viktig terrengparametere, som "specific catchment area" og "wetness index" (Hugget og Cheesman 2002). Man har vært tidlig ute med å ta i bruk GIS. Allerede i 1984 ble det publisert en artikkel om bruk av flytalgoritmer i en rasterbasert grid og de kalte algoritmen sin D8 (O'Callaghan og Mark 1984). Siden den gang har det blitt forslått mange forskjellige flyalgoritmer og de har blitt inkorporert i mange programvarepakker. Men etter hvert har de også blitt tatt i bruk i regional skredanalyse. I Norge har de blitt brukt på steinskred (Romstad et al. 2006), på steinsprang i Alpene (Meissl 1998), på debris flow i Sveits (Huggel et al. 2003b) og i Peru (Huggel et al. 2003a).

Siden både algoritme og datagrunnlag er generaliseringer og forenklinger av virkeligheten, kan deres validitet alltid trekkes i tvil. Hvis opphavet til en terrengmodell eller koden til en flyalgoritme er ukjent, har man mindre mulighet til å vurdere gyldigheten av resultatene. Alle deler av prosessen bør være kjent, slik at resultatene kan reproduseres. Ellers kan det oppstå situasjoner der man er nødt til å trekke sine egne resultater i tvil, fordi man ikke har innsyn i verktøyet han bruker (Ramberg 2007). Da er verktøyet en såkalt "black box", hvor man gir input og får en output, uten noe innsyn. Også koden i flytalgoritmer har, av varierende årsaker, ikke blitt offentliggjort (Tarboton 1997).

Siden betydningen av skredforskningen er så viktig er det betimelig å ta metodene som brukes i nærmere øyesyn. Hvilke effekter får implementeringen av algoritmen på resultatene? Hvilke kritiske valg må gjøres underveis? Hvis forskjellene på de praktiske effektene av valgene blir trivielle i den skalaen en regional analyse opererer, blir de kun av teoretisk interesse. Hvis ikke må de dokumenteres, slik at man kan vurdere analysens validitet på bakgrunn av dem.

1.2 Problemstilling og hypoteser

I denne oppgaven er målet å se nærmere på bruk av flytalgoritmer til å simulere steinskred, hvilke premisser som må legges til grunn for en slik simulering, og hvilke problemer som kan oppstå. Det legges til grunn at man har kjennskap til topografiske data og skal gjøre analysen i en regional kontekst og at man allerede, ved hjelp av andre metoder, kjenner skredets løsneområde.

En modifisering av en skredalgoritme som er velkjent i litteraturen benyttes. Denne algoritmen (D3) vil anvendes på en gridbasert terrengmodell av området rundt Geirangerfjorden og åtte andre utvalgte steder i Norge hvor det foreligger reelle skreddata. Følgende hypotese fremsettes:

”i en regional kontekst og med kun tilgang på topografiske data, kan en skredsimulering ved bruk av D3-algoritmen gi anvendbare resultater.”

Denne hypotesen testes ved å implementere D3-algoritmen, og teste algoritmen mot reelle skreddata. Spesielt fokus rettes mot hvordan algoritmen takler forskjellige egenskaper ved terrengmodellen. Ved å vise i detalj hvordan algoritmen reagerer på terrengmodellen og hvordan den fungerer sammenlignet med D8-algoritmen, evalueres algoritmens egnethet.

Et aspekt som ofte er neglisjert i forbindelse med flytalgoritmer, er håndteringen av uavgjorte resultater ved valg av flytretning (eng. ties). Hyppigheten så vel som konsekvensen, er aspekter som bør belyses. En følgehypotese for dette:

”forekomsten av ties kan være signifikant i gridbaserte terrengmodeller, men forekomsten av og konsekvensen av disse får sjelden effekt for resultatet av D3-algoritmen.”

Denne hypotesen testes ved å analysere terrengmodeller med forskjellig opphav og varierende geografisk plassering. Konsekvensen av eventuelle ties vil så bli gransket med D3-algoritmen, og en utvidelse av denne kalt D3R-algoritmen.

Det eksisterer mange forskjellige algoritmer for flytsimulering. Disse har i stor grad økende kompleksitet. Derfor er en annen hypotese utformet:

Denne hypotesen testes ved å utføre kvalitative sammenligninger av simuleringer ved bruk av den D3-algoritmen med en algoritme som tillater 360 graders variasjon av skredretninger (D360-algoritmen). Simuleringene vil sammenligne algoritmene med hverandre på kunstige flater, forskjellige terrengmodeller og med reelle skred. Dette vil vise om det er hensiktsmessig å til å benytte algoritmer som medfører med kompleksitet.



I kapittel to vil teorien gjennomgås. Her blir de begrepene som relaterer til testing av hypotesene og til emnet flytalgoritmer belyst. Spesielt vekt blir det lagt på D8-algoritmen, siden den danner grunnlaget for D3. Andre flytalgoritmer det kunne være naturlig å bruke i skredsammenheng blir beskrevet, siden disse vil være relevante til hypotese tre om bruk av mer kompliserte algoritmer. I metodekapittelet blir D3-algoritmen beskrevet, samt utvidelsen for å teste konsekvensen av ties, D3R, beskrevet. Et avbruddskriterium basert på informasjon i teoridelen er implementert. I tillegg blir D8 og D360, som brukes som

sammenligningsgrunnlag, beskrevet. Hvilke grunnlagsdata som brukes og metoden med å sammenligne algoritmer og analysen av ties i en terrengmodell blir beskrevet. I resultatkapittelet presenteres sammenligningene først mellom D3 og virkelige skred og mellom D3 og D8. Siden vil resultatene fra en analyse av forekomsten av ties for Geirangerfjorden og Jostedalsbreen vises, både på terrengmodeller fra ASTER-sensoren og Statens Kartverk. D3R blir brukt for å analysere konsekvensen. I den tredje delen sammenlignes D3 med D360 og virkelige skred for å finne ut om D360 gir en mer treffsikker simulering og dermed på om det er noe å vinne ved å bruke mer komplekse algoritmer, enten D360 eller noen av de andre nevnt i teoridelen. Også her brukes terrengmodeller fra ASTER-sensoren og Statens Kartverk.

De tilnærmingene som gjøres i forbindelse med bruk av skredmodeller og skredalgoritmer diskuteres kort i fjerde del av kapitlet. Til slutt gis en konklusjon og oppsummering, og aktuelle undersøkelser videre skisseres.

1.4 Begreper

I denne oppgaven brukes flere algoritmenavn hyppig. Det presiseres her at begrepene D3, D3R og D360 refererer til algoritmer som er implementert i forbindelse med denne oppgaven. De beskrives nærmere i metodekapittelet. D8 refererer til en kjent algoritme. Denne algoritmen er også implementert i forbindelse med denne oppgaven, men kan i noen detaljer skille seg fra andre implementasjoner. Metodekapittelet inneholder en dokumentasjon på hvordan D8 er implementert i forbindelse med denne oppgaven.

Begrepene algoritme, metode og modell brukes også ofte. Med en algoritme menes de matematiske og logiske stegene som er samlet i en prosedyre. Men mange vil referere til sin fremgangsmåte for å simulere skred som en metode. Siden en simulering innebærer en generalisering og abstraksjon av virkeligheten, brukes også begrepet modell. I en modell kan både algoritmer og metoder inngå. Der begrepet skredmodell brukes, referere det til den helhetlige tilnærmingen som gjøres, mens skredalgoritme refererer til de konkrete operasjonene som inngår som en del i en slik modell.

Fire begreper som finnes i engelsk litteratur er *debris flow*, *tie*, *pit* og *sink*. Disse navnene brukes i oppgaven fordi oversettelse vil føre til begrepsforvirring, og de blir definert i teoridelen.

2 Teori

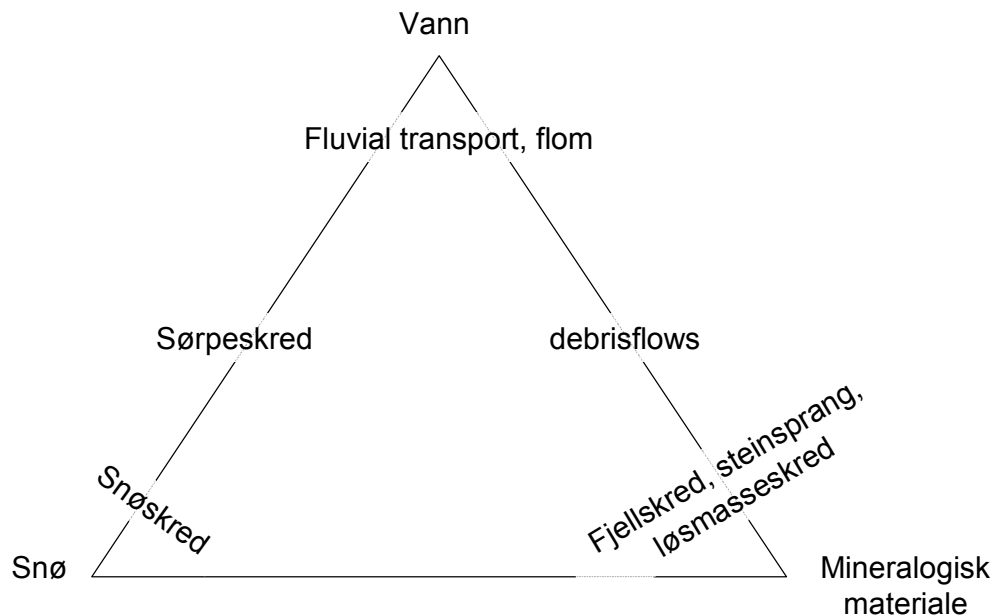
I dette kapittelet blir først steinskred som fenomen undersøkt. Siden D8-algoritmen har vært brukt på andre skredtyper, og erfaringer derfra benyttes i denne oppgaven, settes steinskred også i sammenheng med andre typer skred. Flytalgoritmene skal benyttes i en regional analyse og på en rasterterrengmodell, så disse emnene belyses. Videre nevnes kort andre typer terrengmodeller, som har en viss relasjon til andre flytalgoritmer. Et rammeverk for skredmodeller generelt presenteres, og elementer fra både prosessbaserte og empiriske modeller nevnes, siden dette relaterer til valg av avbruddskriterium og en eventuell utvidelse av D3-algoritmen. Forskjellige flytalgoritmer beskrives, da disse som regel er utviklet som følge av begrensninger i D8-algoritmen. Egenskaper ved disse belyser problematikken rundt D3 og D8 og gir en pekepinn på hvor relevante de er i forhold til skredsimulering.

2.1 Steinskred

Blikra et al. definerer steinskred på følgende måte ”*We define rock avalanches as gravitational mass movements involving a large mass of rock debris that slides, flows, or falls rapidly down a mountain slope.*” (2004). Steinskred kan i prinsippet forekomme overalt hvor fjellsiden er brattere enn fjellets friksjonsvinkel. Forskjellige årsaker kan utløse et skred (Blikra 2003, Braathen et al. 2004), og når skredet først er utløst beveger det seg nedover i terrenget så lenge gravitasjonskraftens komponent i bevegelsesretningen overstiger friksjonskraften. Steinskred kan klassifiseres i tre grupper i henhold til volum: Steinsprang (100 m^3 eller mindre), steinskred ($100 \text{ m}^3 - 10\,000 \text{ m}^3$) og fjellskred ($10\,000 \text{ m}^3$ og større). Til sammenligning er skredene i Loen i 1936 og Tafjord 1934 anslått til å ha et volum på 1 million m^3 (Hestnes 1985a). Andre klassifiseringsmetoder tar høyde for utløsende faktor eller andre egenskaper (Turner og Schuster 1996, Braathen et al. 2004).

Alle typer skred er en hurtig massebevegelse (i motsetning til massebevegelser som kryp og solifluksjon). Figur 2 viser hvordan de relaterer til hverandre. Snøskred og fluvial transport faller utenfor rammene av denne oppgaven. Fokuset er på steinskred, men ettersom flytalgoritmer ved flere anledninger har vært anvendt på debris flow (Iverson og Schilling 1998, Huggel et al. 2003b), blir erfaringer herfra også tatt med. Det er verdt å merke seg aksene mellom massetypene mineralogisk materiale og vann; når man beveger seg fra førstnevnte

mot sistnevnte, kan man på generelt grunnlag anta at viskositeten til massen minker, og dermed massens evne til lateral spredning. Dette har relevans til valg av simuleringsalgoritme.



Figur 2: Klassifikasjon av hurtig massebevegelse (Hestnes 1979)

2.2 Regional analyse

For å analysere steinskred på en regional skala trenger man bruk av dataverktøy som kan automatisere prosessen. Huggel et al. definerer regional skala til å være 50-10 000 km² (Huggel et al. 2003b). Det er nødvendig med en terrengmodell for landskapet i hele regionen. Fra denne terrengmodellen kan forskjellige terrengparametere beregnes, slik som skåning (eng. slope) og aspekt (eng. aspect). Andre kilder kan gi informasjon om andre relevante data som bergrunn og løsmasstype. Skredsonkart er utarbeidet for deler av Norge og kan bidra til å gi generalisert informasjon innenfor et område (Hestnes 1985b). Men ofte kan ikke slik informasjon fremskaffes innenfor de rammer som det kreves for en regional analyse.

En regional analyse har tre hovedkomponenter; en metode som finner løsneområder, en metode som finner skredets bane eller utbredelsesområde, og en metode for å finne skredets lengde (Dorren 2003). En slik analyse kan så inngå i et større metodeverk for å analysere forskjellige farenomenter, hvor forskjellige typer utløsende årsaker, forskjellige typer skred og forskjellige typer konsekvenser, som tsunamilignende kjempebølger håndteres. Slik kan man oppnå et verktøy som for eksempel kan gi råd til utbyggere og administrative organer

(Hestnes 1985a). Et slik rammeverk kan involvere et bredt spekter av teknikker og datakilder, slik som det som er utviklet i Sveits under Hazneth (Metternicht et al. 2005).

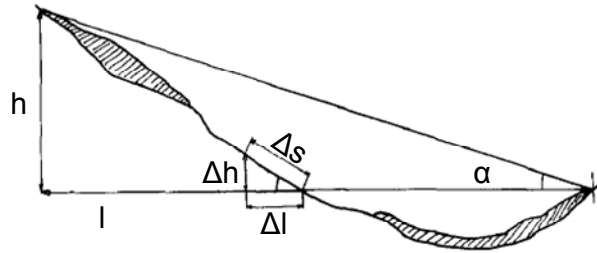
2.3 Skredmodellering

Skredmodellering kan deles inn i tre typer modeller: empiriske modeller, prosessbaserte modeller og topografiske modeller (Dorren 2003). Empiriske modeller tar utgangspunkt i tilgjengelige data og prøver å generalisere en empirisk sammenheng, som ikke nødvendigvis er dypere begrunnet. Det kan for eksempel være sammenhengen mellom volmet til et skred og arealet som er dekket av skredmasser (Dorren 2003). Prosessbaserte modeller ønsker å simulere bevegelse ved hjelp av kjente fysiske lover, mens topografiske modeller foregår ved hjelp av GIS-verktøy, og som oftest på en gridbasert terrengmodell.

2.3.1 Empiriske modeller

En empirisk eller statistisk modell ser på observerte data om skred og ser om det er noen sammenhenger mellom data. En fjellside er en svært kompleks overflate, med mange former og forskjellige underlag hele veien. Skal man simulere en massebevegelse nøyaktig, kan kartleggingen av parametere og beregningsprosessen bli svært krevende, spesielt hvis man vil gjøre det nøyaktig. Ved steinsprang kan man simulere bevegelsen og ta hensyn til luftmotstand, rotasjon, støt osv. Hvis det er en skredmasse man skal simulere, må man på et eller annet nivå gjøre en tilnærming. Istedenfor å bygge opp modelleringen fra grunnen av med alle fysiske parametere, kan man se etter empiriske sammenhenger i kartlagte data. Slike empiriske sammenhenger kan muligens ikke forklares fullt ut under gjeldende forståelse av fenomenet, men er likevel av nytteverdi hvis de har forutsigelsessevne.

Friksjonen er en størrelse som varierer kontinuerlig langs hele skredbanen i forhold til skredets hastighet, underlaget og terrengets helning lokalet. Men i et idealisert tilfellet er friksjonen lokalt lik tangens til den lokale helningsvinkelen, eller forholdet mellom høyden og lengden til den lokale skråningen (Figur 3). Forholdet mellom den totale helningen og den totale lengden er i så måte den gjennomsnittelige friksjonen (Scheidegger 1973). Det som er verdt å merke seg er at bevegelsen er uavhengig av massen til skredet.



Figur 3: Forholdet mellom høyde og lengde (Scheidegger 1973)

Hvis en masse m beveger seg en distanse s , og massen i utgangspunktet var i ro, er den kinetiske energien til gjenstanden lik differansen av potensiell energi mellom startstedet og målestedet.

$$\frac{1}{2}mv^2 = mg\Delta h \quad (\text{Formel 1})$$

Δh er vertikalkomponenten til bevegelsen. Denne størrelsen kan uttrykkes ved bevegelsens lengde Δs langs skråplanet, og β , som er den lokale helningen:

$$\frac{1}{2}mv^2 = mg(\Delta s \sin \beta - f\Delta s \cos \beta) \quad (\text{Formel 2})$$

Nå kan vi omforme til:

$$\frac{1}{g} \Delta \left(\frac{1}{2}v^2 \right) = \Delta h - f\Delta l \quad (\text{Formel 3})$$

For å få for hele skredet, integreres uttrykket over, under forutsetning at skredet starter og slutter i ro. α er den totale helningsvinkelen:

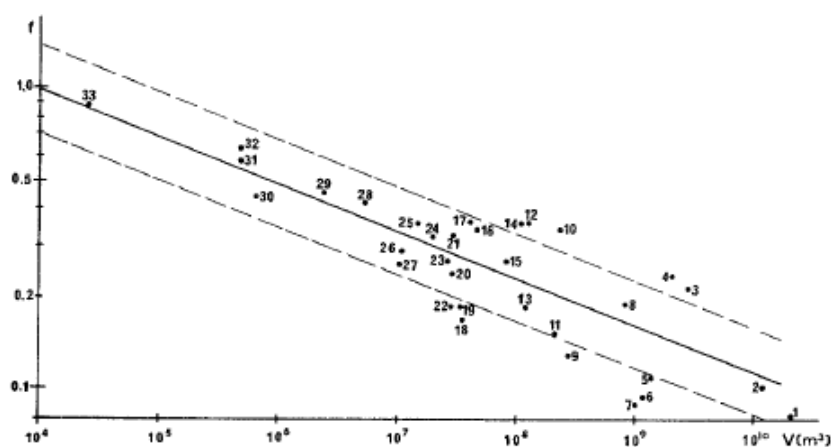
$$0 = h - fl$$

$$f = \frac{h}{l} = \tan \alpha \quad (\text{Formel 4})$$

Denne observasjonen gav en mulighet til å sette opp statistikker for friksjon på skred. Formelen skissert ovenfor er en fysisk lovmessighet. Ved å registrere data for skred kom man over en empirisk lovmessighet; at friksjonen eller h/l er korrelert med inverse logaritmen av volumet (Figur 4). Dette gjelder for volum større enn 100000 m^3 . Under denne grensen holder h/l seg stort sett konstant (Scheidegger 1973).

Dette betyr at hvis man estimerer volumet kan man sette grense i modellen for utløpslengden ved en gitt høyde h . Eller man kan se på det historisk største skredet i området og sette en h/l grense som et verst tenkelig scenario. Disse betraktningene har blitt brukt på andre skredtyper, slik som isskred og debris flow. For debris flow i alpine, settes $\tan \alpha = 0.19$ (Huggel et al. 2003b), mens 0.17 blir foreslått for løsmasseskred i Norge (Hestnes 1985a). Isskred blir i alpine satt til 0.31 (Salzmann et al. 2004). Scheideggers resultater holder stikk også i Norge (Romstad et al. 2006). Steinsprang blir satt til 0.36 for maksimal grense ved steinsprang i Norge (Hestnes 1985a). Disse verdiene er fornuftige når man tar i betraktning hvor tyntflytende debris flow er i forhold til is og steinskred. For steinsprang er en høyere h/l riktig, fordi det er mye mindre masse. Hvis man i et område skal ta hensyn til skred av størrelsesordenen i Tafjord, tilsier det en $h/l = 0.2$.

I mange tilfeller, spesielt i tilfelle ved steinsprang skiller man mellom skredvinkelen og *skyggevinkele*n (eng. shadow angle). Skyggevinkele n er linjen fra der skredet slutter til der steinspranget treffer ura ved et vertikalt fall fra en klippe. Et slikt vertikalt fall kan feilaktige verdier (Jaboyedoff og Labiouse 2003).



Figur 4: Sammenheng mellom h/l og volum (Scheidegger 1973)

2.3.2 Prosessbaserte algoritmer

Prosessbaserte eller dynamiske algoritmer trenger mange parametere for å simulere et skred. En steinblokk i bevegelse nedover en fjellside kan både gli, sprette og rulle og opplever friksjon både fra lufta og underlaget. Et skred inneholder mange individuelle blokker som også utøver krefter på hverandre. Siden dette blir uoverskuelig, må man generalisere kreftene som virker inn. Man kan enten angripe problemet som en energibetraktning, hvor skredets potensielle energi i løsnøyeblikk tilsvarer den kinetiske energien. Ettersom skredet beveger seg nedover reduseres den kinetiske energien og dermed også farten, inntil all skredet stopper og all kinetisk energi er gått over til andre former. Eller man kan se på skredet som en kropp under påvirkning av krefter, tyngdekraften og friksjonskraften. Når tyngdekraftens komponent i fartsretningen overstiger friksjonskraften, øker farten, og vice versa.

Både hastighet og utløpsdistanse avhenger av potensiell energi ($E=mg\Delta h$). I startpunktet er all energien lik potensiell og i endepunktet for skredbanen er alt omgjort til kinetisk energi. Underveis er totalenergi lik potensiell pluss kinetisk energi. Dette gir formelen på hastighet (hvor massen er irrelevant):

$$V = \sqrt{2gh} \quad (\text{Formel 5})$$

Mesteparten av den kinetiske energien omformes dog til varme som følge av friksjon. Hovedpoenget er at farten avhenger av energien som er til overs etter at friksjonen er overkommet. I et gitt punkt vil en skredblokk ha farten:

$$V = f\sqrt{2gh} \quad (\text{Formel 6})$$

hvor f er en størrelse som representerer blokkens rotasjon (Jaboyedoff 2003).

I lokale skredanalyser kan man bruke forskjellige modeller med mange parametere, både statistiske/empiriske og dynamiske/prosessorienterte.

Et eksempel på en fysisk basert modell er STONE, som også er interessant fordi den genererer en TRN overflate (*triangular regular network*) ut ifra en rasterterrengmodell, og bruker det som beregningsgrunnlag (Guzzettia et al. 2002). Andre modeller er PCM, NIS og VSG (Harbitz 2002). Flere eksempler finnes hos Dorren (2003).

2.3.3 Topografiske modeller

I en topografisk modell inngår en terrengmodell og algoritme som beskriver skredets bevegelse i denne, kalt en flytalgoritme. En empirisk eller prosessbasert modell må supplere modellen (Dorren 2003). Dette er fordi terrengmodellene og de avledede terrengparametrene kun beskriver terrengets beskaffenhet eller form og ikke prosessene som foregår i det. Ethvert geografisk fenomen er en kombinasjon av form og prosess (Etzelmüller og Sulebak 2000). De empiriske og prosessbaserte modellene kan, på hver sin måte, tilføre informasjon om prosessen. Dette kan være løsneområdet. Hvis en man ønsker å beregne løsneområdet på bakgrunn av skråning, trenger man likevel informasjon om grenseverdien (for eksempel 45° helning) som skal til for å utløse skred. Og man kan si noe om forholdet mellom utløpsdistansen og høyden innenfor en topografisk modell, men man trenger likevel data som sier noe hva dette forholdet skal være større enn før skredet stopper.

2.4 Terrengmodelltyper

2.4.1 Raster

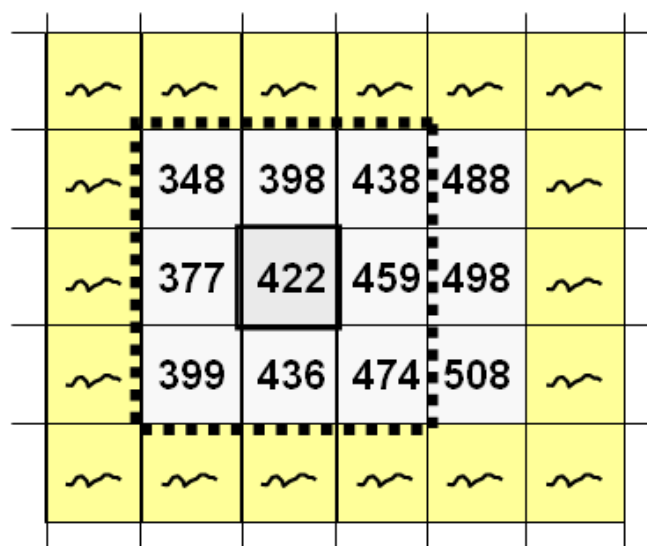
Den vanligste benyttede terrengmodellen i skredanalyse, er et raster (Figur 7b). Hele det geografiske området er diskretisert i like store, kvadratiske enheter, som kalles celler og hvor høyden eller et annet aktuelt attributt er uniform innfor hele enheten. Oppløsningen til rasteret er størrelsen på disse cellene, og presisjonen er usikkerheten til et gitt terrengattributt. Rasteret er en tilnærming til en kontinuerlig flate, høyden varierer ikke kontinuerlig, slik som terrenget i virkeligheten gjør, for ved hver cellegrense er det sprang til en annen verdi. Dette betyr at modellen ikke er kontinuerlig, og det må tas hensyn til ved beregninger på modellen.

Et GIS-verktøy kan utføre mange forskjellige analyser på en raster. Disse analysene grupperes gjerne etter hvor mange celler som inngår i en enkelanalyse, og kan kombineres til mer avanserte analyser. Flere avanserte analyser kan kombineres til en modell. En analyse som ser på hvert enkelt punkt, kalles punktanalyse, mens en analyse som inkluderer alle nabocellene rundt ett punkt, kalles en fokalanalyse, analyser på grupper av punkter som alle er forbundet kalles soneanalyse (eng. zonal analysis), mens en analyse på hele rasteret kalles global analyse (Burrough og McDonnell 1998). Hver av disse analysene utføres ved å løpe gjennom alle punkter eller soner i rasteret.

En raster er blitt svært populært som terrengmodeller fordi de lagres effektivt i datamaskinens lagringsstrukturer, kan underkastes avanserte analyser med GIS-verktøy og kan kombineres med letthet med satellittdata som er lagret i rasterformat. Disse dataene kan supplere terrengmodeller, danne grunnlag for dem der de ikke ellers er tilgjengelig, gi informasjon om terrengtype og brukes til overvåkning, fordi de er basert på kontinuerlig overflyvende satellitter og ikke omfattende flybildeserier.

Selv om et raster er velegnet for datamaskinens lagringsstruktur, er det et lite fleksibelt format. Alle deler av terrenget opptar like mye lagringsplass, uansett om det er en slette eller en kompleks struktur. Dessuten forsvinner detaljer mindre enn cellestørrelse og skarpe overganger i terrenget blir overdrevet. Men disse problemene er blitt mindre med tiden (Wilson og Gallant 2000).

Et fokalt vindu er som oftest 3x3 celler (Figur 5). Dette brukes svært ofte i bildeanalyse, i diverse filtre, hvor verdien i hvert celle utjevnes av verdiene i alle nabocellene. For eksempel utføres et gjennomsnitt, noe som har som effekt å redusere variasjonen til verdiene i bildet. Vinduet kalles et bevegelig vindu, fordi det beveger seg som et vindu gjennom gridet, og ser på hver celle etter tur. Flytalgoritmer baserer seg på slike vinduer, hvor neste retning avgjøres på forskjellige måter.



Figur 5: Bevegelig vindu (Berry 2008)

I rasterterrengmodeller forekommer det ofte *sinks* og *pits* (Figur 6). I en pit har en flyt ikke noe utløp, da alle nabocellene ligger høyere. En sink er et område med flere slike pitceller, uten noe utløp. Slike områder kan oppstå som følge av feil i terrengmodellen eller være genuine forekomster i terrenget. En pit for eksempel kan skyldes for eksempel åpninger i tredekke, enslige trær eller støy (Fairfield og Leymarie 1991). Fordi de stopper en flytsimulering ønsker man å fjerne dem, og det kan gjøres ved hjelp av fyll-algoritme. I en slik sammenheng er det alltid en fare for at man fjerner genuine terrengformasjoner. Derfor kan man regulerer algoritmen med å sette en maks fyllhøyde.

103	59	84
89	50	71
75	59	61

Figur 6. Eksempel på en pit som utgjøres av en enkeltcelle

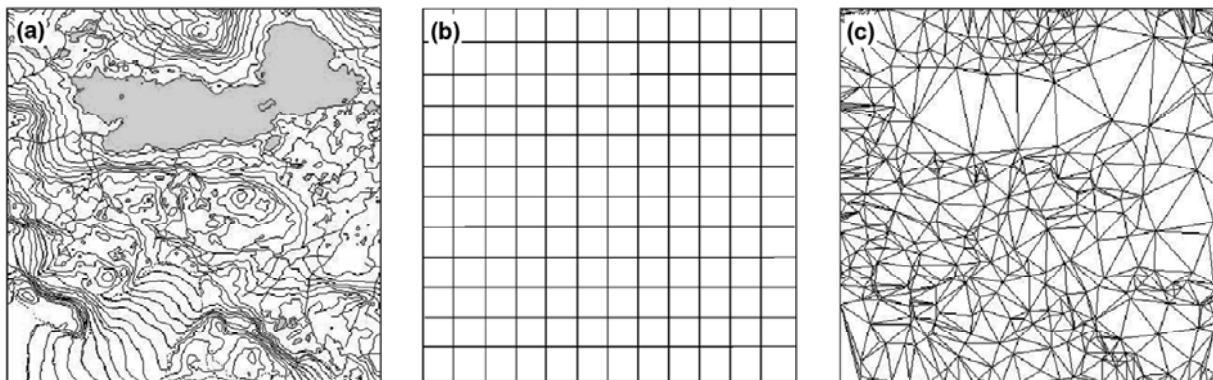
I en skredsammenheng forholder det seg annerledes. Man er interessert i å fjerne små pits med utbredelse på en eller noen få celler. Mest sannsynlig er de unøyaktighet i terrengmodellen. I andre tilfeller hvor de tilsvarer ekte terrengformasjoner, kan man likevel si at et skred vil ha så stor hastighet at det vil passere over slike mindre formasjoner. Mange skred har vist hvordan de faktisk kan gå oppover på andre siden av dalen. I slike sammenhenger er det liten tvil om at en fyll-algoritme bare vil styrke modellen, siden algoritmen ikke tillater skred å gå oppover. Når det er snakk om en større sink kan det være at terrenget vil bremse skredet så mye at det stopper. Fylles en slik sink, nødvendiggjør det et avbruddskriterium som tar hensyn til bremseeffekten ved store flater. Et slik avbruddskriterium kan være en gitt h/l verdi.

2.4.2 TIN

TIN (*triangular irregular network*) er en alternativ terrengmodell, der alle flater er diskretisert inn i trekanter som beskriver terrenget (Figur 7c). Denne lagres på en mindre intuitiv måte i datamaskinen, men er mye mer fleksibel en et raster. Der hvor det er mange detaljer, er datatettheten høy, mens flate områder krever lite data for å beskrives.

2.4.3 Kontur

En konturmodell består av høydelinjer, hvor alle punktene langs en høydelinje har samme høyde. Et gitt intervall mellom høydelinjene, kalt ekvidistansen gjør at avstanden mellom kotelinjene er konstant. I mellomrommet mellom kotelinjene antas terrenget å varierer konstant, og kan dermed interpoleres. Konturbaserte terrengmodeller danner ofte grunnlaget for rasterbaserte terrengmodeller. Det finnes flytalgoritmer som benytter denne modellen (Fairfield og Leymarie 1991).



Figur 7: Terrengmodeller (a) kontur, (b) raster, (c) TIN (Etzelmüller og Sulebak 2000)

2.5 Flytalgoritmer

Flytalgoritmer defineres av Huggel et al. som ”*algorithms for transferring flow sequentially to lower points or areas*” (Huggel et al. 2003b). De er opprinnelig brukt innenfor hydrologien, hvor man er opptatt av å simulere drenering og flom. Et vesenlig begrep er *Specific Catchment Area* (SCA)¹, hvor man er opptatt av hvor stort areal som drenerer gjennom et bestemt punkt. Med andre ord spør etter alle celler som på et gitt tidspunkt drenerer gjennom et gitt punkt. Vet man dette kan man si noe om hvor mye nedbørsvann, eller sedimenter eller forurensning som kommer til å drenerer gjennom dette punktet ved å anta en gitt nedbørsmengde for hvert celle. Eller man kan si noe om forurensningen til punktet på en lignende måte (Hugget og Cheesman 2002, Tarboton 2003). For vite dette for et gitt punkt må man dermed først spørre om hvilken retning hver celle i griddet drenerer i, og så gå til det gitt punktet man vil analysere og nøste seg bakover. Det er derfor et grid som er laget av en

¹ *Upslope area* defineres som det totale dreneringsfelt over et gitt punkt eller del av en konturlinje SCA defineres som *Upslope area* delt på konturavstanden (Burrough og McDonnell 1998).

flytalgoritme ofte kalles et "backlinkgrid". Det finnes metoder for å rekursivt jobbe seg bakover (Tarboton 1997).

Innenfor skredsimuleringen er man ikke opptatt av SCA. Derfor er en del av de betraktningene som gjøres i litteraturen på validiteten til forskjellige algoritmer ikke nødvendigvis gyldige i kontekst av skredmodellering. For eksempel beskriver Tarboton hvordan en skredberegning bør være deterministisk og ikke preget av spredning, fordi beregningen av et SCA bør kunne gjentas på lik måte (1997).

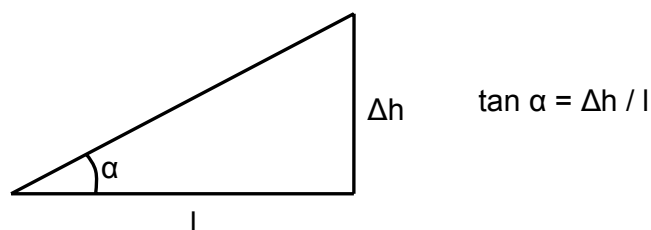
Andre skredtyper enn steinskred kan også simuleres med flytalgoritmer. Huggel et al. bruker både en modifisert singelflytalgoritme og en multipleflytalgoritme til å simulere debris flow. En debris flow er en skredtype som befinner seg mellom flomskred og løsmasseskred og har ofte et svært høyt vanninnhold. Dermed har det lavere viskositet, og må forventes å ha en større lateral spredning på flatt terreng enn for eksempel steinskred. Analysen viser hvordan flytalgoritmer kan tilpasses ikke bare skredsimulering, men også til forskjellige skredtyper etter som hvilke egenskaper ved skredet man ønsker å fremheve (Huggel et al. 2003b).

En annen modell som er en kombinasjon av en GIS-modell og empirisk modell er LAHARZ. Den modellerer debris flow slik som MSF (se nedenfor), eller mer spesifikt laharer. Denne bruker D8 til å beregne flyten og har et h/l forhold som avbruddskriterium. I tillegg utvider den D8-algoritmen ved å modellere hvor høyt opp langs sidekantene skredflommen når (Iverson og Schilling 1998). Noen flytalgoritmer er basert på konturmodeller og istedenfor raster, slik som Moore sin (Fairfield og Leymarie 1991).

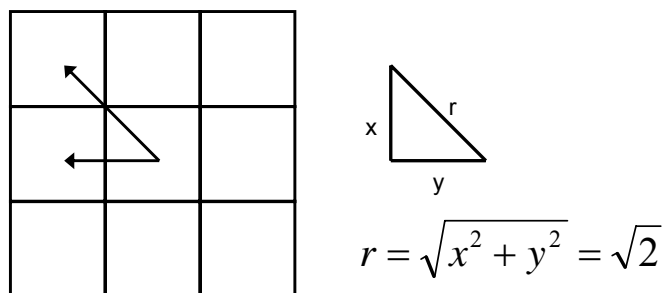
2.5.1 D8

Den første flytalgoritmen som ble tatt i bruk kalles D8 (deterministisk i åtte retninger) (O'Callaghan og Mark 1984), og er fortsatt utbredt. Den er implementert i for eksempel ArcInfo (ESRI 2007), TAU-DEM (TAUDEM 2008) og Tapes-G (Wilson og Gallant 2000). Et bevegelig vindu undersøker naboene til alle celler eller celler i et grid. Hvis ingen av nabocellene har en lavere høydeverdi enn sentercellen, markeres cellen som en pit. Hvis ikke cellen er en pit, finner algoritmen den bratteste retningen. Høydeforskjellene mellom sentercellene og alle nabocellene gjennomgås. Brattheten til en skråning er gitt ved vinkelen på skråningen. Tangens til helningsvinkelen er lik høydeforskjellen delt på avstanden fra sentrum av sentercellen til sentrum av nabocellen (Figur 8). Lengden settes her til en, som en

grunnenhet, siden alle celler i griddet er av samme størrelse. I de diagonal retningene vektet retningen med roten av to, i henhold til Pytagoras. Dette kalles distansevektning (Figur 9). Siden cellene kun skal sammenlignes innbyrdes, er det akseptabelt å finne den bratteste skråningen ved å sammenligne høydeforskjellene. Men hvis brattheten av en eller annen grunn skulle sammenlignes med vinkler fra andre grid, må man kalkulere med cellens utstrekning.



Figur 8: Helningsvinkel α



Figur 9: Vekting i diagonale retninger

D8 undersøker åtte naboer, sammenligner høydeforskjellen og velger en av disse som neste skritt. Tidlige varianter inkluderte å kun se på høydeforskjellen og ikke helningsvinkelen. Dvs. at man utelot distansevektning. Man kan også velge å undersøke bare fire naboer (NESW), istedenfor åtte. Goodchild bruker denne metoden. Disse metodene vil føre til at mulig avvik på 44° . ”To the best of our knowledge, all published algorithms save that of Goodchild [1988], mentioned above, uses method D8” (Fairfield og Leymarie 1991). Ved å bruke fire retninger oppnår man ikke å annet enn å forsterke grid-effekten, og ved å utelate distansevektning gjør man gir man de diagonale retningene en kunstig fordel.

En annen variant er å tillate flyt til alle åtte naboene, men basere kalkulasjonen på de fire kardinale nabocellene. Denne variantene sammenligner ikke bare brattheten til neste celle *ut ifra* sentercellen, men tar hensyn til alle disse fire cellene når den kalkulerer skråningen. Dette er parallelt til plantilpasningsalgoritmene beskrevet nedenfor og benytter de nevnte nabocellene til å interpolere den lokale flaten. Sagt på en annen måte, D8 tar ikke hensyn til

naboceller som er høyere enn sentercellen, men denne varianten gjør det (Wilson og Gallant 2000). Denne varianten skal gi mer nøyaktige resultater, men kan heftes med at høye naboceller kan gi et resultat som avviker fra den bratteste flytretningen (Wilson og Gallant 2000).

D8 sjekker de åtte hovedretningene etter tur. Men det er ikke veldokumentert og velkjent hva som skjer hvis to eller flere helningsvinkler er like. Slike uavgjorte tilfeller kalles ties. Resultatet avhenger da av hvilken rekkefølge algoritmen sjekker i, fordi den første av to like blir valgt. Det er overlatt til måten algoritmen er implementert i de forskjellige softwareløsninger.

D8-algoritmen velger kun en retning for flyten, og det er av dens største begrensninger. Dette er en oppførsel får konsekvenser på flatt terreng og på konvekse flater (Tarboton 1997). I slike situasjoner forventer man ofte en lateral spredning. Selv om steinskred har mer moment og en skredmasse som er mindre viskøs enn debris flow og vannflommer, har også de lateral spredning.

2.5.2 Grid-effekt i D8

En annen av bakdelene ved D8 er av flyten kun kan bevege seg i en av åtte hovedretninger med 45° mellomrom. Den reelle helningsvinkelen i punktet, aspektet, kan ha en helt annen retning. Hvis aspektet ligger langs en av de kardinale eller diagonal hovedretningene, vil flaten samsvare med aspektet. Men hvis aspektet ligger i noen av mellomrommene kan avviket bli på så mye som 22,5°. Dette omtales i som grid-effekt (eng. grid bias) eller anisotropisk (retningsavhengig) flyt og er veldokumentert både for idealiserte flater og virkelige terrengmodeller (Fairfield og Leymarie 1991, Costa-Cabral og Burges 1994, Tarboton 1997, 2003). Det reduserer heller ikke på grid-effekt å øke oppløsningen med griddet (Fairfield og Leymarie 1991).

Hvis det er stor variasjon i terrengmodellen relativt til oppløsningen har grid-effektene mindre å si. Men hvis aspektet varierer mindre enn 45° innenfor en avstand på 10 celler, vil man få signifikante feil med D8 (Fairfield og Leymarie 1991). Med andre ord, med høy oppløsning på terrengmodellen eller svakt bølgende landskap øker grid-effekten. Det er dessuten vist at mengden støy som skal til for å gjøre D8 isotropisk (uavhengig av retning), varierer både med

helningsvinkelen og aspektet til helningen (Fairfield og Leymarie 1991). Dette at oppløsningen forstreker problemet, bekreftes av Tarbotons som viser at forskjellene i virkelige terrengmodeller med 30x30m oppløsning er trivielle, mens for 2x2m er de betydelige (Tarboton 1997).

Grid-effekter viser seg i virkelige terrengmodeller også som parallelle baner (Wilson og Gallant 2000). Dette er baner som ut ifra terrenget burde konvergere, men fortsetter i parallelle, uavhengige løp. O'Callahan og Mark, som først foreslo D8, kommenterer at *"the parallel channels which arise on uniform slopes may be undesirable for some applications"* (Fairfield og Leymarie 1991). Mange av de alternative algoritmene som er foreslått (se under), omhandler strategier for å håndtere grid-effekt. Det er også foreslått å introdusere tilfeldige feil i terrengmodellen for å overkomme dette problemet (Wilson og Gallant 2000).

2.5.3 D16

Denne algoritmen minner mye om D8, men tar i bruk et større vindu istedenfor (5x5). Denne metoden har vært brukt til å simulere steinsprang, både lokalt og regionalt sammen med en empirisk modell for utløpsdistanse (Meissl 1998). Grid-effekt er redusert noe, siden man har 16 vinkler å velge mellom, og maksimum forskjell mellom virkelig aspekt og simulert retning er redusert fra 22,5° til 11,3°. Dette går på bekostning av mer prosessering.

850	840	845	855	860	850	840	845	855	860
840	835	837	845	845	840	835	837	845	845
825	830	832	830	835	825	830	832	830	835
815	820	825	825	817	815	820	825	825	817
805	800	810	803	804	805	800	810	803	804

Figur 10: Sammenligning av D8 og D16 (Dorren 2003)

D16 introduserer et større nabolag. Måten dette er gjort på er å se på celler som er lenger vekt, og dermed la algoritmen kunne velge mellom 16 retninger. Dette reduserer grid-effekten, men fjerner det ikke. Dessuten ignoreres de cellene som ligger i mellom sentercellen og det ytre nabolaget. Det er dog forskjell på dette, og det å inkludere ett større nabolag, men fortsatt velge en av nabocellene. Hvis man har et vindu på 5x5 celler, men fortsatt velger en av åtte retninger, har man ikke oppnådd noe (Fairfield og Leymarie 1991).

2.5.4 MSF

Som nevnt tidligere simulerer ikke D8 spredning. I noen skredformer er dette særs relevant. Huggel et. al. tok høyde for dette ved å modifisere funksjonen 'pathdistanse' i Arcinfo (2003b). I denne funksjonen bruker man D8 sammen med gridet som et "kostnadskart" for å finne den raskeste (dvs. bratteste) banen. For å simulere spredning tillater man at algoritmen avviker opptil 45° fra bratteste retningen. Vektingen av avviket øker lineært. Forholdet mellom horisontal avstand og kostnadsavstanden fungerer som en indikator på hvor sannsynlig det er for at en gitt celle blir truffet. En sti som inneholder mye avvik fra bratteste skråningen får en gradvis lavere sannsynlighet. Denne metoden ble kalt MSF (Modified Singel Flow) (Huggel et al. 2003b).

Metoden er topografisk basert og bruker en empirisk basert avbruddsmekanisme hvor h/l er minimum 0.19. Huggel et al. vedgår at metoden verken gir en sannsynlighetsverdi i matematisk forstand eller at den er fysisk fundamentert i fenomenet den simulerer. Like fullt oppnåes gode resultater med å simulere debris flow i de sveitsiske alpene, og denne metoden foretrekkes framfor en metode basert på FD8 algoritmen (Huggel et al. 2003b).

2.5.5 Rho8

D8 algoritmen er deterministisk og hvert punkt avgjøres uavhengig av alle andre. Man kunne hindre grid-effekt ved enten å gjøre hver avgjørelse om retning, avhengig av tidligere avgjørelser. Hvis skråningen heller mot NNW, kunne man tenke seg man først gikk et skritt mot N og så et mot NW etc. Fairfield og Leymarie mener dette introduserer for mye kompleksitet og hevder det er bedre å gjøre algoritmen stokastisk istedenfor (1991). Tanken er at man kan oppnå det samme resultatet over et større område, men uavhengig av hverandre. Ved hvert skritt introduserer man en tilfeldig komponent, som vektes i henhold til estimert

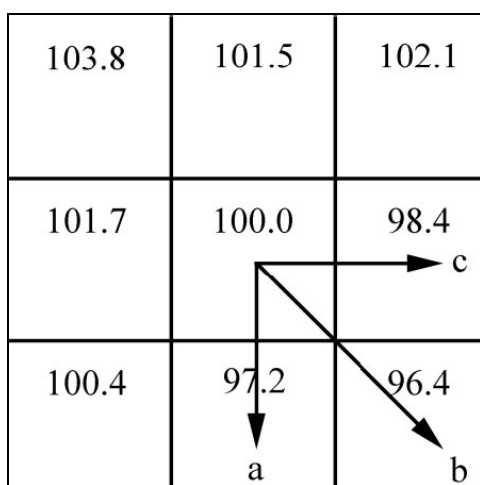
aspekt. Man velger å styre unna mer kompliserte metoder for estimere aspektet, og velger å se på henholdsvis fire eller åtte naboer, derav navnene Rho4 og Rho8. Fordelen med dette er at tilfeldige variable kan få tilfeldige verdier i individuelle tilfeller, men har en forventningsverdi over tid. Fairfield mener at denne forventningsverdien vil sammenfalle med det estimerte aspektet over tid. Med andre ord, hvis aspektet er NNW, velger man å la vært punkt velge mellom N og NW med 50% sannsynlighet for hver. Over et lengre tidsrom vil forventningsverdien bli NNW.

Denne algoritmen er blitt kritisert for å gi svært avvikende resultater, spesielt på korte avstander, sammenlignet med de andre algoritmene nevnt her (Wilson et al. 2007). Tarboton avviser denne fordi den ikke gir reproduserbare resultater, og er ikke konsistent med beregningen av SCA (Tarboton 1997). Det siste argumentet er ikke relevant i skredsammenheng, men det er lett å forestille seg at algoritmen trenger en viss utløpsdistanse for å gi et korrekt anslag. Hvis algoritmen, på en uniform flate, trenger et gitt antall steg for at variasjonen skal falle innenfor et gitt konfidensintervall, kan man stille spørsmål om hvor riktige svarene blir på en overflate som varierer på tilfeldige måter, slik som en virkelig terrengmodell. Algoritmen er en del av TAPES-G (Wilson og Gallant 2000)

2.5.6 FD8

En måte å simulere spredning på er, er å simulere flyt til flere lavereliggende celler. Slike algoritmer kalles gjerne multiflyt-algoritmer eller MF (eng. multiple flow). FD8 er et eksempel på en slik algoritme (Quinn et al. 1991). Den allokere flyt til alle lavereliggende celler, og andelen flyt er proporsjonal med skråningen, dvs. høydeforskjellen delt på en distansevekt i diagonale retninger på en lignende måte som hos D8 (Se Figur 11). Siden algoritmen ikke er begrenset til en retning, er den heller ikke heftet med grid-effekter. Algoritmens tendens til å modellere for sterkt spredning har blitt dokumentert av Tarboton. Den har for sterk spredning på konvekse og plane flater, men akseptabel spredning på konkave flater, for her er spredningen begrenset av at flaten konvergerer (Tarboton 1997). Tarboton mener også spredning er inkonsekvent i forhold til å bruke en flytalgoritme til å beregne SCA, men dette er ikke relevant i en skredsammenheng. Den er brukt til å simulere debris flow, men ble kritisert for ikke å være robust nok (Huggel et al. 2003b). I denne studien

foretrekkes MSF framfor FD8², for MSF hevdes å være mer robust og mindre avhengig av brukerinteraksjon, mens FD8 likevel mindre sensitiv til feil i terrenghmodellen. Wilson et al. viser at FD8 er den algoritmen som skiller seg mest fra de andre flyt-algoritmene de testet (D8, Rho8, DEMON, Dinf og FD8) (Wilson et al. 2007). Algoritmen er tilgjengelig i FORTRAN kode (Huggel et al. 2003b) og som en del av verktøyet TAPES-G (Wilson og Gallant 2000).



Figur 11: FD8 allokterer flyt til alle lavereliggende celler

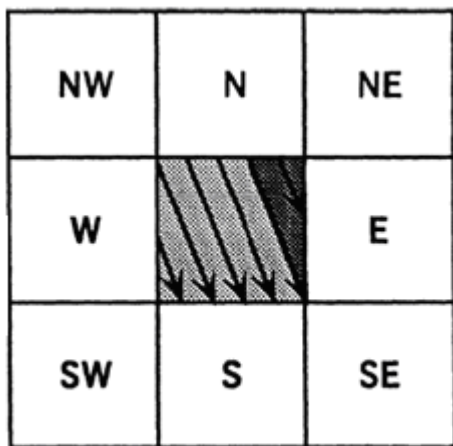
2.5.7 DEMON

To algoritmer baserer seg på å finne flytretningen ved å beregne aspektet. De finner dette ved å tilpasse et flatt plan til hvert celle, og bratteste skråningen er den retningen som en ”rullende ball” ville følge. Den første som algoritmen som brukte dette konseptet var 2D-Lea (Wilson et al. 2007) og algoritmen DEMON bygde videre på ideene i 2D-Lea (Costa-Cabral og Burges 1994). Retningen oppgies i inkremitter på 1°, i motsetning til inkremitter på 45° hos D8. Algoritmene skiller seg fra hverandre på hvordan planet tilpasses hvert celle, og på at DEMON modellerer flyten på sub-cellenivå.

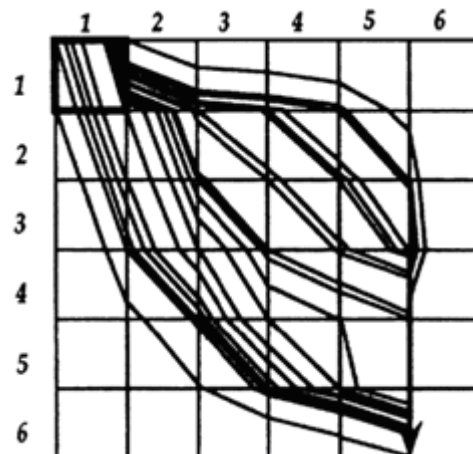
I begge tilfeller tilpasses planet til de fire cellehjørnene, men disse hjørnene representerer forskjellige ting i de to algoritmene. I 2D-Lea representerer de gjennomsnittet av høydeverdien i cellene som grenser til hjørnet, mens DEMON tar høydeverdiene som de er.

² Referert til som MF (Multiple Flow) i artikkelen.

2D-Lea tillater flyt kun til en celle og resulterer i en flyt som er blitt beskrevet som trappeaktig (Tarboton 1997). DEMON tillater flyt til en eller to liggerliggende celler. Hvis flytretningen som treffer cellekanten, følger en av de kardinale hovedretningene, går all flyten til en celle (akkurat som D8), men er flyten diagonal, går den til to kardinale naboer. I Figur 12) **Error! Reference source not found.** deles denne flyten mellom celler i retning S og E. En innvending mot D8 er at flyten modelleres som en endimensjonal størrelse (en linje). Dette er en innvending som kan gjøres mot de fleste flytalgoritmene, siden de ikke tar hensyn til sub-cellenivå, dvs. hva som foregår innenfor et celle. DEMON simulerer flyt som en todimensjonal "flow-tube", hvis bredde avhenger om det lokale planet konvergerer eller divergerer (Figur 13).



Figur 12: Flyt i DEMON



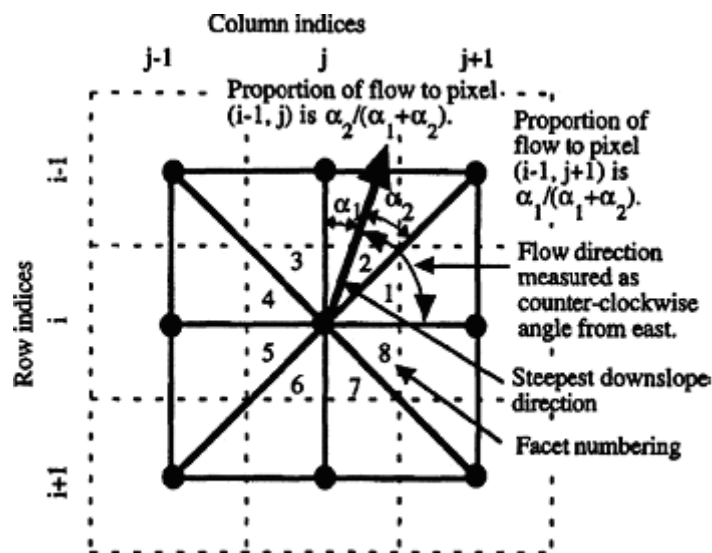
Figur 13: "Flow-tube" i DEMON

Algoritmene kan gi presise resultater, men takler virkelige terrengmodeller dårlig. Tarboton viser hvordan DEMON rapporterer 6607 pits i en terrengmodell hvor alle pits allerede er fylt. Både 2D-Lea og DEMON er lite robuste ovenfor terrengartefakter, for eksempel sal-formede formasjoner, og kan gi en loop-oppførsel mellom to celler (Tarboton 1997). Naboceller med stor høyde gir for stor innflytelse på resultatet, noe som gjør algoritmen mindre velegnet på grovt terreng, noe også Costa-Cabral og Burges innrømmer (Costa-Cabral og Burges 1994). DEMON blir svært komplisert, og må utvides med tilleggskode for å takle spesielle tilfeller som skissert ovenfor.

Det er vist at D8 og DEMON gir mer sprikende resultater enn noe annet algoritme-par (Wilson et al. 2007). DEMON er tilgjengelig i TAPES-G (Wilson og Gallant 2000), men her beregnes aspektet på samme måte som D360 algoritmen (se metodekapittelet).

2.5.8 Dinf

For å bygge videre på Leas metode og DEMON, utviklet Tarboton (1997) Dinf eller D_∞ . Den lar retningen variere kontinuerlig mellom 0 og 2π . Hvis bratteste retning sammenfaller med en av de kardinale eller diagonale retningen, er algoritmen identisk med D8 i at en av nabocellene velges. Hvis ikke tillater algoritmen spredning, men bergrenser spredningen til to celler. Det bevegelige vinduet deles inn i åtte fasetter, og den bratteste finnes ved å sammenligne nedstrømsvektorene innenfor hver av disse. Hvis vektoren faller utenfor fasettets sentrumsvinkel på 45° , settes den lik den nærmeste kanten. Den tillater flyt til en eller to celler.



Figur 14: Flytretningen på triangulære fasettplan (Tarboton 1997)

Algoritmen sjekker fasettet som ligger mellom E og NE først, og fortsetter så mot klokka. Hvis det forekommer en tie, velges den første. Hvis algoritmen treffer på en pit, finnes retningen ved å se på et grid som er preprosessert med D8-algoritmen hvor pit håndtering er implementert (Tarboton 1997, 2003, ESRI 2007). Dette skjer ved å peke på en celle som ikke er en pit, og gjentas iterativt, til man kommer ut av sinken. Algoritmen er tilgjengelig gjennom verktøyet TAUDDEM (TAUDDEM 2008). Studier viser at DEMON og Dinf tenderer til å gi like resultater som forventet, siden begge tillater spredning til en eller to celler, men likeheten er ikke så sterk som forventet på reelle terrengmodeller (Wilson et al. 2007).

3 Metoder

Metodekapittelet består av fem deler. I den første delen beskrives alle grunnlagsdataene, deretter presenteres oppbygningen av D3-algoritmen i detalj. Så blir metoden for å analysere forekomsten og konsekvensen av ties presentert. I fjerde del presenteres algoritmen som D8 skal testes opp mot, D360. Til slutt blir sammenligningene som skal gjøres mellom D3 og D30 skissert.

3.1 Metodedesign

For å teste hypotesene er tre algoritmer implementert; D3 for å teste hypotese en. D8 er implementert for å sammenligne denne. Videre er en Matlab-funksjon implementert for å analysere forekomsten av ties, mens en algoritme, D3R, skal teste konsekvensen. Dette inngår i testingen av hypotese to. En tredje type algoritme, D360, er implementert for å teste hypotese tre. Virkelige skreddata (eng. ground true) er innhentet for å validere resultatene. Sammenligningen mellom algoritmer, mellom algoritmer og skred og analysen av tie-forekomster, er alle kvalitative analyser.

3.2 Grunnlagsdata

3.2.1 Terrengmodell fra Statens Kartverk

Det er benyttet en terrengmodell fra Statens Kartverk, distribuert av Ugland IT Group, og er basert på digitaliserte kartdata. Terrengmodellen har en RMS (*Root Mean Square*) på ± 5 -6 meter. Den har en oppløsning på 25 meter. Terrengmodellen refereres heretter til som SKT25.

Funksjonen "setnull ([DEM] == 0, [DEM])" kjøres på terrengmodellen, for å sette alle med høydeverdi 0 meter til NoData (manglende høydeverdi), med hensikt å gjøre fjorden til en barriere. Algoritmen avslutter hvis den leser inn en celledverdi med NoData. Der det nevnes at terrengmodellen er blitt prosessert med en fyll-algoritme, er det "fill" funksjonen i ArcInfo som er benyttet (ESRI 2007).

3.2.2 ASTER-terrengmodell

Satellittdata er av økende betydning for å generere terrengmodeller. For å generere en terrengmodell basert på data fra en optisk sensor, trengs det stereobildepar. ASTER-sensoren om bord på Terra satellitten var en av de første som registrerte stereopar, fordi den i tillegg til hovedsensoren også har en sensor som peker bakover i banen (*along-track stereo*). Dette gir en kvalitetsforskjell ovenfor stereopar som er generert basert på flere satellittoverflyvinger (*repeat-track stereo*) (Huggel et al. 2003b). Den genererte terrengmodellen har en oppløsning på 30 meter i 70% av modellens utstrekning. Nøyaktigheten er på ± 15 meter RMS. Scenen er registrert 03.08.2002 (Hausberg 2008).

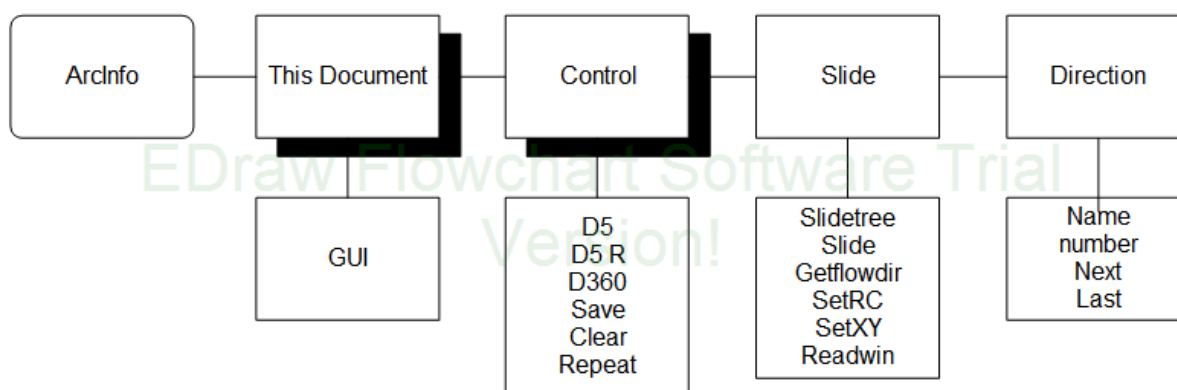
3.2.3 Skreddatabasen

Skreddatabasen er digitalisert av Bård Romstad (UiO) og basert på skisser fra Lars Harald Blikra (NGU). Skredene er enten digitalisert som polygoner eller linjer.

3.3 Implementasjon av D3-algoritmen

For å simulere skredbaner implementeres D3 som er basert på D8. D8 er ment brukt til flytmodellering innenfor hydrologien, men det er fullt mulig å bruke den til skredanalyse. En mulighet er å ta utgangspunkt i ett eller flere løseområder og bruke "flowdirection" i ArcInfo til å finne mulige baner. Siden resultatet av en "flowdirection" er en "backlinkgrid", kan man så bruke funksjonen "costpath" med terrengmodellen som vektgrid til å finne den bratteste banen til et gitt løseområde. Fordelen med dette er at resultatet i høy grad er reproducerbart som følge av ArcInfos utbredelse. Men en ufordel er at det er en lite interaktiv analyse, hvis man raskt vil gjøre forandringer på løseområdet. En viktigere motsigelse mot å bruke den på skred, er at den ikke tar hensyn til Newtons første lov for massebevegelse. Et legeme, i dette tilfellet et skred, som er i bevegelse, har en motstand å forandre bevegelsesretning. Det har et visst *moment*. "Flowdirection" tillater en simulering av bevegelse som gir rom for å "snu på en femøring". Riktignok vil en slik snuoperasjon forekomme innenfor en gitt celle som kan ha en bredde på 25 meter, men det strider mot den intuitive oppfattelsen av en skredbevegelse. Å tillate så brå bevegelser Det er derfor ønskelig å legge inn begrensninger på bevegelsesfriheten.

En annen mangel ved "flow direction" er mangelen på et avbruddskriterium. I en regional analyse trenger man en begrensning på utløpslengden. Ellers vil man få uønskede resultater i forbindelse med grensene for griddet og på flate områder og ved vannoverflaten. For "flowdirection" kan man i mange sammenhenger legge inn barrierer som håndterer de mest opplagte av slike problemer ved å definere uønskede områder som NoData. Man kan sette vannflater og områder med en viss helning, til NoData med funksjoner som "slope" og "reclass". Men helningsgraden lokalt har begrenset verdi som avbruddsparameter. Riktignok kan man anta at de omkringliggende områdene har lignende helning, og at hvis man setter et begrensning på minimumstørrelsen til NoData-områdene, kan man få et brukbart kriterium for barrierer. Men vil man sette et avbruddskriterium som tar hele skredets bane og høydeforskjell i betraktning, må man bruke andre strategier. Som nevnt i teoridelen er det blitt utført regional skredanalyser ved å bruke funksjonen "Pathdistance" med modifikasjoner (Huggel et al. 2003b). "Pathdistance" baserer seg på et kostnadsgrid (eng. cost grid) som i dette tilfellet er terrengmodellen, hvor bratteste skråning er minste motstands vei. Man kan med denne metoden definere et ønsket avbruddskriterium også. En annen strategi har blitt lagt til grunn i denne oppgaven.



Figur 15: Oversikt over programmet

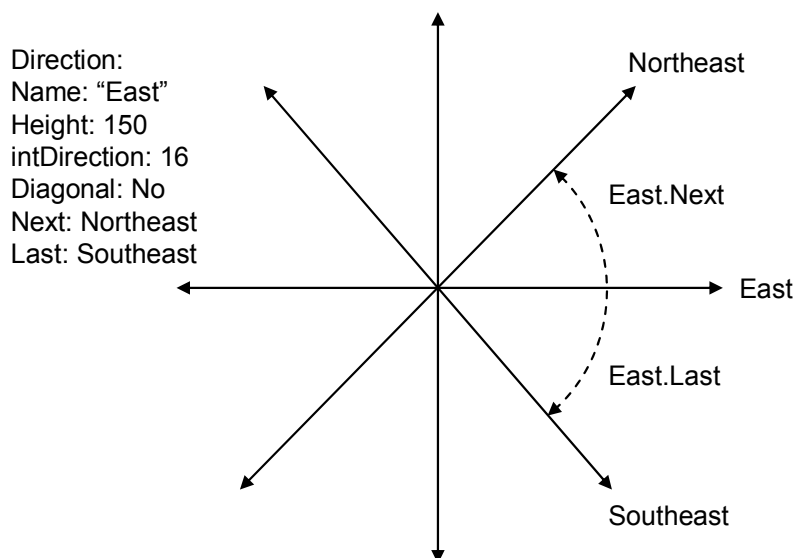
For å få større kontroll over og fleksibilitet i bruken av algoritmen, er den implementert fra bunnen av i ArcInfo ved hjelp Visual Basic for Applications (Microsoft 2008). Algoritmen kjøres som en del av et skript som kan aktiviseres ved å velge et punkt i terrengmodellen manuelt. Skriptet kjører algoritmen som velger neste celle, fra startcellen og på hver utvalgt celle deretter. Dette medfører også en betydelig reduksjon i prosesseringen, siden

vindusoperasjonen kun utføres på cellene i banen, ikke i hele gridet. Figur 15 viser hovedstrukturen til skriptet.

Det er klassen skred som håndterer simuleringen av et skred, mens *Control* kommuniser med et brukergrensesnitt ovenfor brukeren i ArcInfo. Rutinen *slide* mottar informasjon om punktet og beregner skredet steg for steg gjennom gjentatt kall til rutinen *getflowdir*. Siden beregning for hvert steg er en fokal operasjon, må et vindu med høydeverdier leses inn fra rasteret for hvert steg. *Slide* er også navnet på klassen som samler alle disse funksjonene. Hver gang et skred skal analyseres lages det et *slide-objekt*, og for hver celle som analyseres er det åtte *retningsobjekter*. VBA er et objektorientert språk (Dahl og Nygaard 1966) og objekter er velegnet til å håndtere mer komplekse entiteter slik som retningsobjektene (Burrough og McDonnell 1998).

3.3.1 Det bevegelige vinduet

Det bevegelige vinduet er en todimensjonal array på 3x3 verdier, som leses inn fra rasteret basert på den lokale gridkoordinaten. Verdiene leses inn for hvert steg algoritmen tar nedover stien. Hver av disse verdiene representerer en høyde, men også en retning algoritmen kan ta.



Figur 16: Implementering av retning i D3

Algoritmen skal ikke sjekke alle retninger, men kun i samme retning som sist og innenfor et tillat avvik av denne. trengs det en kobling mellom celleverdier og retninger. Med et tillat avvik på 45° og forrige retning som nordvest (celle-koordinat (0, 2)), skal retning nord (1, 2) og vest (0, 1) også sjekkes. Istedenfor en kobling mellom disse cellene, er det enklere og mer intuitivt å assosiere disse cellene med et retningsobjekt som peker på naboene sine (Figur 16).

Hvis dette er på plass er det mye enklere implementasjonsmessig å unngå å referere feil celleverdi i vinduet. Dette gjør det også enklere å implementere eventuelt nye funksjoner basert på vinduet, som skråning og aspekt.

ression	Value	Type	Context
bDiagonal	True	Boolean	Slide.getflowdir
intDirection	128	Integer	Slide.getflowdir
name	"southeast"	String	Slide.getflowdir
pLast		Direction/Direction	Slide.getflowdir
pNext		Direction/Direction	Slide.getflowdir
z	1103	Double	Slide.getflowdir
z	1098	Double	Slide.getflowdir
z	1095	Double	Slide.getflowdir
z	1094	Double	Slide.getflowdir
z	1091	Double	Slide.getflowdir
z	1094	Double	Slide.getflowdir
z	1098	Double	Slide.getflowdir
z	1101	Double	Slide.getflowdir

Figur 17: Eksempel på retningsverdier for en gitt celleblokk

Det letter også feil-finning, når man ved kjøring av skriptet kan ha oversikt over alle celleverdiene og forsikre seg om at den riktige høyden blir valgt (Figur 17).

For hvert punkt i banen kalles en rutine som inneholder utvelgesesalgoritmen. Den kalles *getflowdir* og gangen er skissert i Figur 18. Rutinen må vite forrige retning som ble valgt, for å kunne sette en begrensning for hvilke algoritmer som er gyldige. Hvis for eksempel NE var forrige gyldige retning, er NE, E og N gyldige retninger.

Startpunktet er et spesialtilfelle. Uansett begrensning på avvik fra forrige retning, må alle retninger være gyldige. Hvis retningen ble satt vilkårlig og man bare tillot sjekk i tre retninger, ville man risikere å få rapportert en pit startpunktet i skråninger hellende mot vest, hvis startretningen ble vilkårlig satt mot øst.

Algoritmen finner bratteste helning. Følgende formel ligger til grunn (Burrough og McDonnell 1998, Wilson og Gallant 2000):

$$[\delta x / \delta z]_{ij} = \max[(z_{i+1,j} - z_{i-1,j}) / 2] / dx \quad (\text{Formel 7})$$

δx er her avstanden mellom sellesentrene. Den antas her til å være lik for alle cellene i gridet, og brukes under opptelling av avstanden flyten har tilbakelagt. Så det er kun høydeforskjellen som måles. De diagonale retningene vektes i henhold til Pytagoras med kvadratroten av 2.

Et avbruddskriterium ligger innebygd her. Hvis noen av cellene som sjekkes inneholder NoData, rapporteres punktet som en pit, og algoritmen avsluttes. Dette er strengt tatt ikke korrekt, og en eventuell utvidelse ville levert en egen kode for NoData (for eksempel -1), hvis det er ønskelig å skille mellom disse. Det viktigste er dog at algoritmen avslutter ved møte med barrierer, slik som gridkant og definerte NoData-områder, slik som havoverflaten (høyde 0). "flowdirection" har mulighet til å velge om flyten skal ignorere kanten (eng. edge contamination), eller drenere ned fra den. Dette er ikke prioritert her.

Subrutinen returnere en tallkode for den bratteste retningen. Denne gis på formen 2^n , med verdiene (1, 2, 4, 8, 16, 32, 64, 128). Hvis flere gyldige retninger detekteres, er det ingen kombinasjoner av retningsverdier som vil tilsvare noen annen retningsverdi.

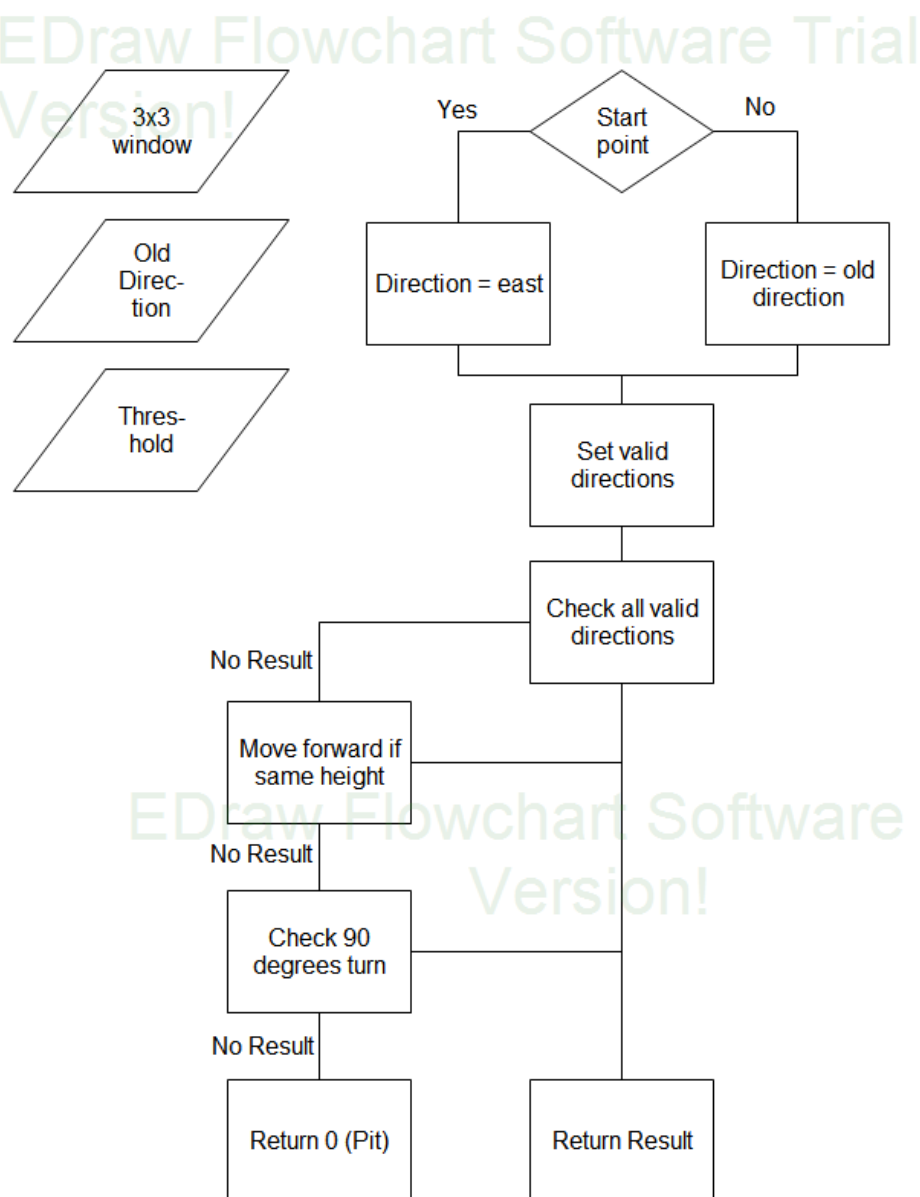
1	2	4
8		16
32	64	128

Figur 18: Retningsverdier i D3

Det bør bemerkes at resultatet logisk sett aldri kan bli 255. Det er kun startpunktet som kan være en spiss, altså at sentercellen har den største høyden. Algoritmen ellers ville ikke bevege seg inn i et slikt punkt. Hadde algoritmen tillat flyt inn i en celle ved lik høydeverdi i alle retninger, kunne dette oppstått, hvis flyten kom inn på en flate.

Vel å merke er denne kodingen av retninger tilfeldig, og andre måter er vanlige. Arcinfo bruker en rekkefølge som går mot klokka og hvor øst er en, mens TAPES-G setter nordøst til en (Wilson og Gallant 2000).

Hvis ingen gyldige retninger finnes, kan algoritmen tillate at flyten fortsetter forover. Dette er av samme årsak som begrensningene på avvik i vinkel; skredet har et visst moment. Det tillates ikke å fortsette i et lateralt avvik på 45° , hvis høyden er lik. Hvis man lot algoritmen gjøre det, ville det ha den fordel at det reduserte sjansen for at flyten stopper opp ytterligere, men kan gi uønskede effekter på flatt, konvekst terreng.



Figur 19: Flytdiagram for valg av retning i D3

En ytterligere utvidelse kunne være å sjekke ytterligere retninger. Valget er tatt om at algoritmen tillater avvik fra forrige retning i 45° graders vinkel, men dette kunne økes til 90 grader. Som nevnt tidligere, er dette intuitivt ikke kompatibelt med oppfattelsen om skred som massebevegelse. Men det har den fordel at det unngår uønskede stoppsituasjoner, hvor en mulig sti kun finnes ved en dreining på 90°. I regionale analyser er det ofte et behov til å begynne med å åpne for et "worst-case" scenario. Under den forutsetning kan det argumenteres for at man burde åpne for avvik mot 90 grader, for å unngå å miste noen tilfeller.

Følgende pseudokode viser logikken i algoritmen:

```
Function Getflowdir (previous direction)
if startpoint then
    valid directions = all direction
else
    valid directions is old direction and directions with
maxdegree of old direction

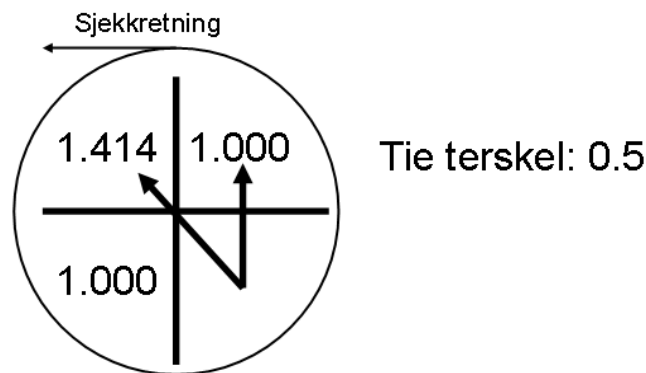
: main loop
    for each valid direction
        weight = 1
        if direction.z = nodata Then Return 0 (Pit) and exit
        if direction.z > center.z then
            if direction is diagonal then weight = square(2)
            difference = (center.z - direction.z) / weight
            if difference - maxdifference >= 0 then
                if difference-maxdifference>=threshold then
                    maxdifference = difference
                    flowdirection = direction
                else
                    maxdifference = difference
                    flowdirection=flowdirection+direction
            direction = direction.next (mot klokka)
        end main loop
        if flowdirection = 0 and not start then
            if previous.direction.z = center.z then
                flowdirection = direction
        Optional:
        if flowdirection = 0 and not start and maxdegree = 45 then
            valid direction = 90 degree
            goto main loop
    Return flowdirection (=[0, 254])
```

Et viktig valg som kommer inn her er håndteringen av ties. I utgangspunktet kan man være interessert kun å finne retningen med bratteste gyldige skråning, og la algoritmen velge denne. Men sett at det forekommer to eller flere gyldige retninger med lik høyde. Hvis kun en retning returneres, og den bratteste blir valgt, blir det brått viktig hvilken rekkefølge retningene sjekkes. I algoritmen ovenfor sjekkes retningen mot klokka. Ved startpunktet begynner algoritmen å sjekke først mot øst, men det er irrelevant, siden alle retninger sjekkes. Hvis antallet ties er forsvinnende lite, eller effekten av dem blir svært liten, er dette med rekkefølge på retningssjekkene noe man kan ta lett på. Men en terrengmodell er en tilnærming av virkeligheten, og hvis presisjonen ikke er svært høy, er det ikke urimelig at ties kan forekomme. Implementasjonsmessig må man ta hensyn til antall gjeldende sifre på variablene i slike sammenhenger. Hvis roten av to lagres i en variabel av *double* format (Microsoft 2008), vil antall gjeldende siffer være så stort at sannsynligheten for ties vil bli forsvinnende liten. Hvis høydeverdiene i terrengmodellen oppgis for eksempel i integer format, må resultatet av roten av to rundes ned til samme presisjon.

Istedenfor å la algoritmen rapportere den første den kommer til av flere like, kan man rapportere alle slik som det gjøres her. Man kan også utvide dette konseptet. Man snakker i mange sammenhenger om uskarpe objekter (eng. *fuzzy objects*) innenfor geografisk informasjonsvitenskap (Burrough og McDonnell 1998). Dette med ties er et eksempel på en størrelse som ikke har noen klar avgrensning. Det kan være mye rett i at helt like høyder forekommer sjelden i naturen, men tilnærmet likhet er derimot svært vanlig. Det er ikke vanskelig å tenke seg situasjoner, som selv om utfallet er deterministisk, er det avhengig av uoversiktlige og uforutsigbare variable. Og hvis man følger filosofien om å åpne for "worst case", bør ikke slike retninger utelukkes fra en analyse. Man kan tenke seg en divergerende sti ved en tie som følger to helt forskjellige løp. Man vil miste mye viktig informasjon ved å utelukke slike tilfeller fra analysen. Derfor er det i algoritmen ovenfor innført en terskelverdi (eng. *threshold*) hvor det tillates en vist slingringsmonn for ties. Den bratteste retningen fortsatt den som det sjekkes opp mot, men nå godtas alle skråninger med helningsvinkel innfor slingringsmonnet satt av terskelverdien.

Det er en svakhet ved håndteringen av ties. Hvis det forekommer tre ties (eller flere, hvis man tillater at det sjekkes i flere enn tre retninger), kan man det oppstå en situasjon hvor den første verdien faller utenfor slingringsmonnet til den siste som sjekkes, men blir likevel regnet som ties, fordi den første er innefor slingringsmonnet til den andre, som er innenfor

slingringsmonnet til den tredje. I Figur 20 er terskelverdien 0,5. Det betyr at de to første verdiene settes som ties. Men den tredje regnes ikke som tie, selv om denne er innenfor terskelverdien. Den nordvestlige retningen har økt maksverdien, og siden $1,4141 > 1,000$, registreres ikke dette som tie. Algoritmen bør derfor utvides til å gå igjennom gyldige retninger to ganger. Men dette sannsynligvis forekomme ytterst sjelden; to forhold må gjelde: det må være en trippel (eller mer) tie og retningene må fordele seg etter et vist mønster.



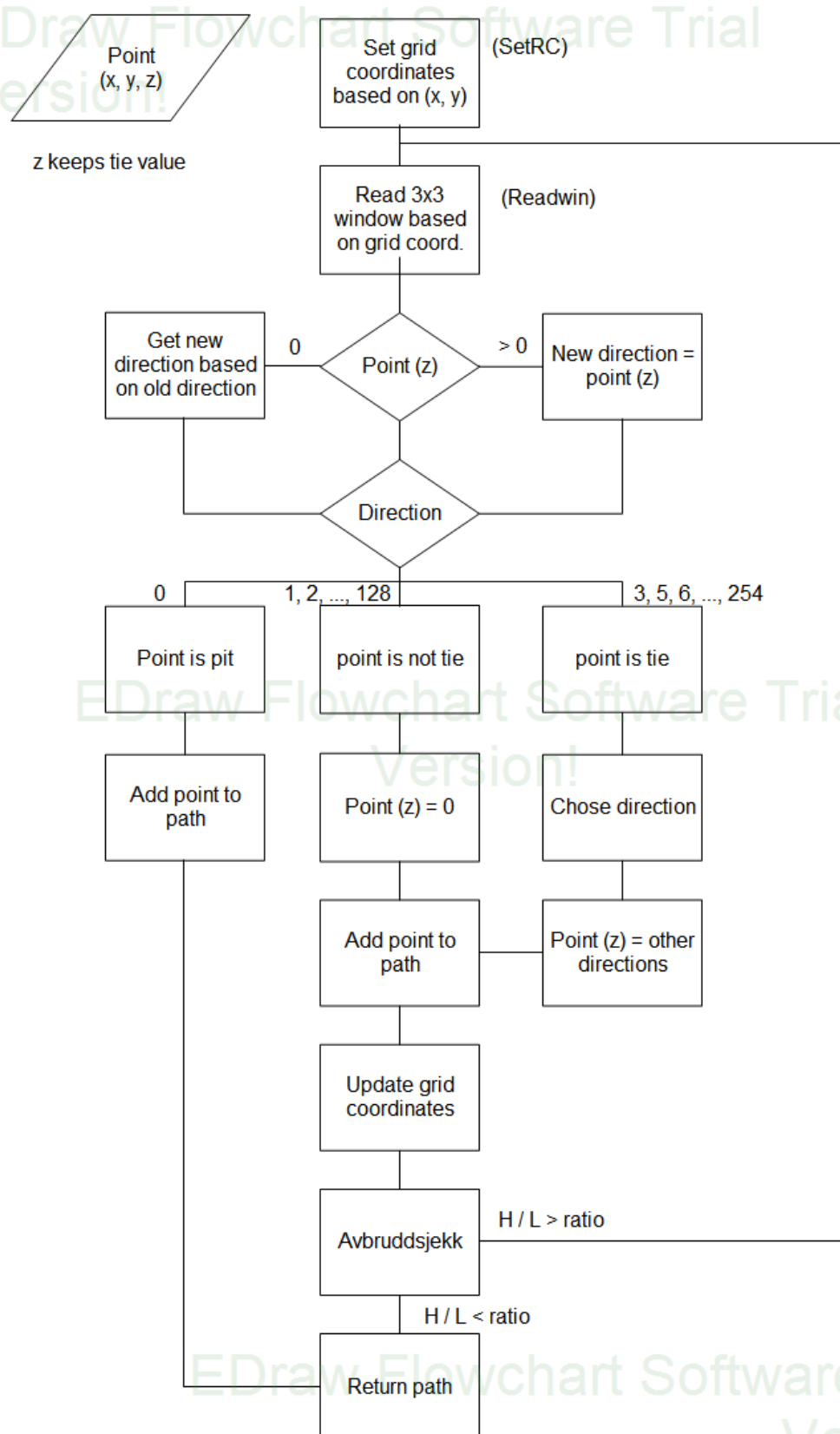
Figur 20: Begrensning i algoritmen

Det er enkelt å implementere D8-algoritmen som en utvidelse av metodeverket over. Enhver kall av rutinen *getflowdir* som gir neste retning, må gjøres med beskjed om at dette er første punkt. Dermed vil alltid alle retninger sjekkes. Dette er implementert for å gi en mulighet til å sammenligne med D3-algoritmen.

3.3.2 Simulering av skredbanen og avbruddskriterium

Rutinen *slide* utgjør rammen rundt utvelgelsen av neste celle (**Error! Reference source not found.**). Det er den som finner hele banen, og holder styr på posisjonen i griddet og leser inn vindusverdiene. Det er også her eventuelle avbruddskriterier som tar hensyn til mer enn akkurat det bevegelige vinduet må ligge.

Den enkleste form for avbruddskriterium er å fange opp områder som er definert som NoData. Dette gjøres i rutinen *getflowdir*. I *slide* kan man også legge inn et enkelt stoppkriterium for antall skritt. Dette kan i de enkleste tilfelle være tilstrekkelig. I analysen rundt en fjord, kan man anta at alle løsneområder har skred som enten ender i en sink eller i fjorden. På den måten overser man ingen resultater. Eller man kan hvis man har historiske skreddata, lage en maks utløpslengde for regionen.



Figur 21: Rutinen Slide i D3 som håndterer individuelle skred

Mer sofistikert er det dog å legge inn et avbruddskriterium basert på skrittets totale lengde og høydeforskjell. Man tar forholdet mellom høydeforskjellen mellom et gitt celle og startpunktet, og den opptelte lengden. Lengden inkrementeres med cellebredden for hvert skritt og distansevektes med roten av to. For hvert skritt sjekkes h/l mot en grenseverdi.

H/l ble satt til 0,02 for de følgende analysene. 0,02 er det som ble funnet for skred i Norge på 10^6 m^3 (Romstad et al. 2006). Det er ikke et mål for oppgaven å finne en ideell h/l verdi. Den

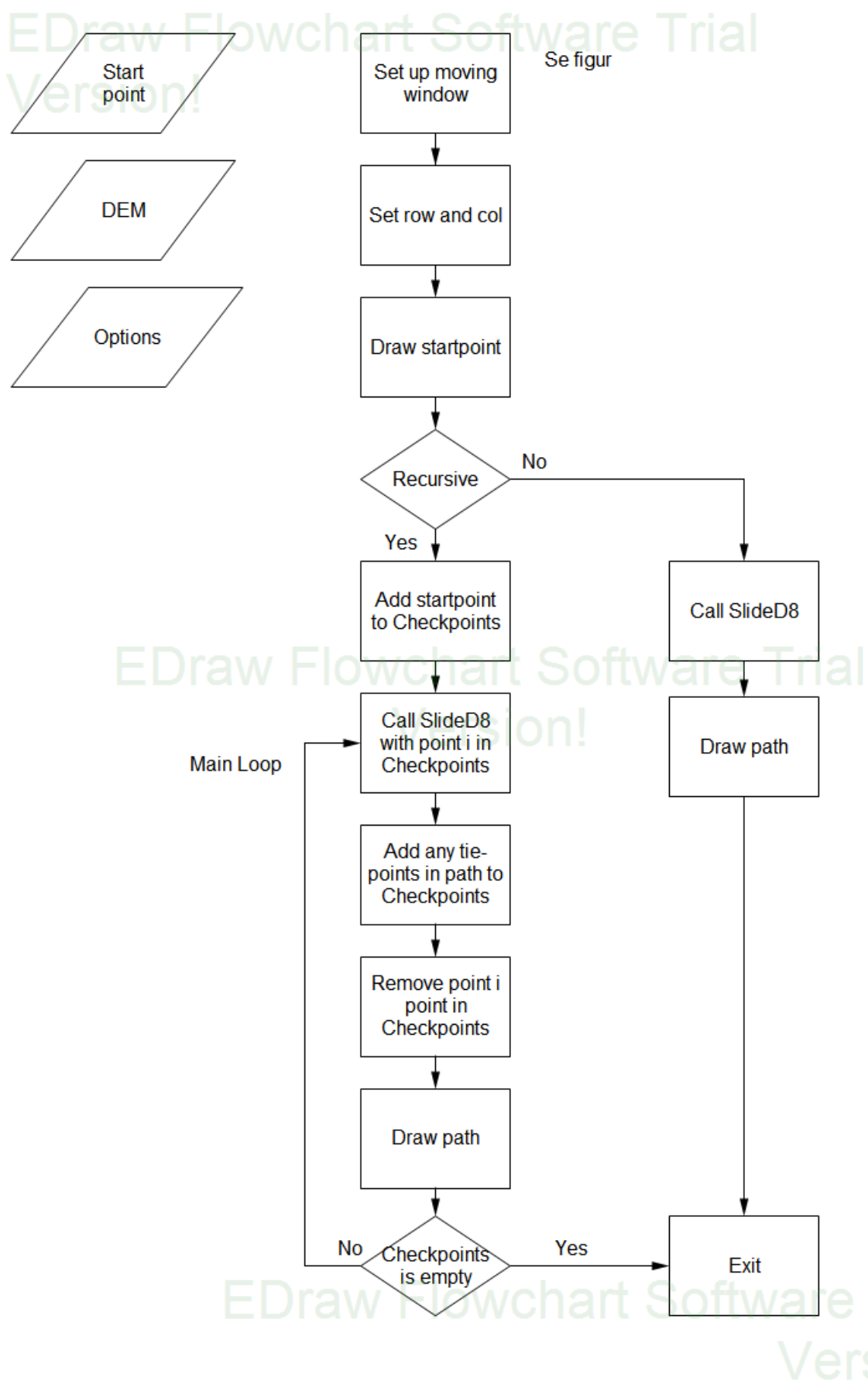
må settes basert på empiriske data fra den regionen som skal undersøkes. Den valgte verdien er ment som et ekstremresultat, en absolutt grense for hva som er mulig, ikke en realistisk simulering som tar i betraktning de lokale forholdene på et gitt sted.

Hvis verdien som *getflowdir* finner, er en sum av flere retninger, velges den høyeste retningsverdien først. Den resterende verdien lagres i punktet som legges inn i samlingen punkter som utgjør banen. Neste gang programmet kommer til dette punktet, kalles ikke *getflowdir*, fordi de resterende retningene som skal sjekkes, er allerede definert. Siden programmet henter inn celleverdiene til vinduet for hvert steg, lagres retningsverdien i punktets z-variabel, som ellers er forebeholdt høydeverdien.

3.4 Håndtering av ties

3.4.1 Implementering av D3R

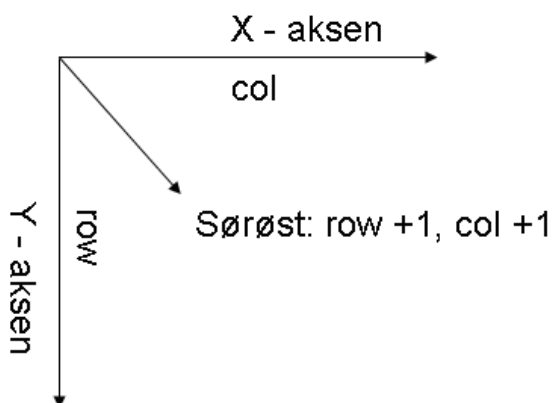
Rutinen *slidetree* ble implementert for å gi mulighet for å implementere D3 rekursivt og derav navnet, siden skredsimuleringen da får en struktur som minner om en hierarkisk trestruktur (Figur 24). *Slide*-rutinen markerer de punktene som den finner med tie-verdier, men gjør ikke noe med dem. *Slidetree* undersøker den banen som *slide* har funnet og tar vare på eventuelle tie-punkter før den tegner opp banen. *Slide* kalles så på nytt igjen med det første av eventuelle slike punkter. Den behandler dermed alle punktene som om det var helt nytt brukervalgt startpunkt. På grunn av dette gir *slidetree* sliderutinen beskjed om punktet den får er et startpunkt, siden dette skal behandles særskilt. Se Figur 22 for oversikt over metoden.



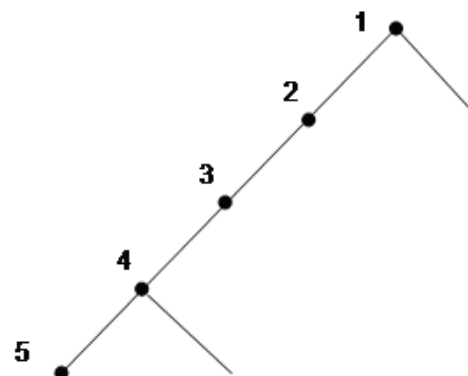
Figur 22: Rutinen slidetree håndterer tie-punkter

Inputene til algoritmen er rasterterrengmodellen. Data fra denne modellen blir blant annet brukt til å sette opp de interne variablene *row* og *col* som er algoritmens måte å holde styr på posisjonen på. Et koordinatsystem med *row* og *col* har origo i øvre venstre hjørne. Det betyr at hvis algoritmen finner at neste retning er sørøstover, må *row* inkrementeres med 1 og *col* inkrementeres med 1 (Figur 23). Det er en vanlig konvensjon er å bruke øvre venstre hjørne, spesielt i bildeprosessering og med en y-akse pekende nedover, men positiv y-akse forekommer også (Wilson og Gallant 2000).

Det bevegelige vinduet settes også opp i *slidetree*. Hvis algoritmen ikke er rekursiv, tegnes kun en bane opp og algoritmen avsluttes. Hvis ikke går programmet inn i en løkke, hvor hver sti som beregnes, gjennomgås etter tie-punkter. Disse sjekkes i sin tur, til alle punkter er undersøkt (Figur 24).



Figur 23: Lokalt koordinatsystem



Figur 24: tie-punkter (1 og 4)

En begrensning i algoritmen er at alle punkter som er markert som ties, kalles som om de var et nytt startpunkt. For det første fører dette til at overlappende baner, i fall, to stier fra et tie-punkt konvergerer, tegnes opp på nytt. Skulle det forekomme svært mange ties, vil dette føre til at beregningen tar tid, og i beste fall stopper programmet. Et forsøk på å håndtere dette ved hjelp en liste over besøkte punkter ble forsøkt (se appendiks). Men optimalisering faller utenfor rammene av denne oppgaven. En annen begrensning er at det er vanskeligere å implementere avbruddskriterier i D3R. Alle verdier, som tilbakelagte steg bør lagres, for eksempel i et objekt, sammen med selve punktet.

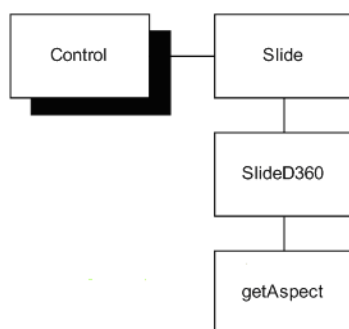
3.4.2 Analyse av forekomst av ties

Det blir gjort en analyse i programmet Matlab (Mathworks 2008) på utvalgte områder fra SKT, som Geirangerfjorden og Jostedalsbreen. Ved hjelp av et skript laget av Bård Romstad (UiO) analyseres flytretningene for hvert punkt (se Appendiks). Dette vil gi både et visuelt overblikk over fordelingen av ties og hvor mange ties som finnes for et gitt tie-punkt, dvs. om det er to, tre eller flere like bratte retninger for et gitt punkt. Her vil en grenseverdi avgjøre hvor like to helninger må være for å være en tie. Hvis det viser seg at tie-punkter forekommer sjelden, er det mindre viktig å ta hensyn til dem ved bruk av D8-lignende algoritmer.

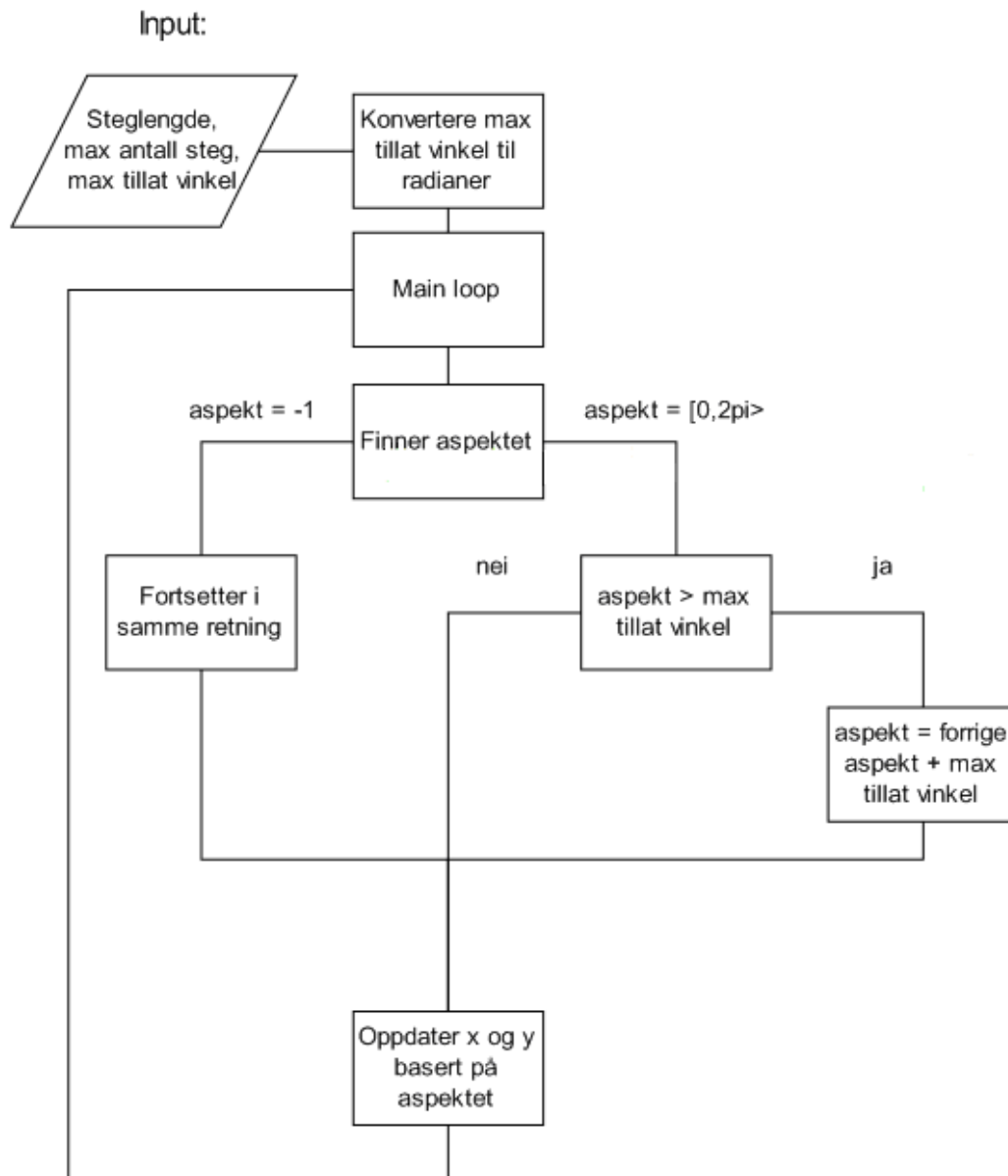
Det er forventet at resultatet blir dominert av to områder: Sinks vil nødvendigvis vises som områder med flyt i alle retninger, siden algoritmen tillater flyt videre ved lik høyde, og av områder med bare en gyldig retning. Dette vil gi en indikator på hvor vanlig ties er. En analyse av to forskjellige terrengmodeller for Jostedalsbreen, SKT og ASTER, vil vise hvordan ties fordeles i terrengmodeller med forskjellig opprinnelse. D3R-algoritmen brukes til å analysere konsekvensen av ties på Geiranger terrengmodellen og sammenligne med reelle skred (eng. ground truth) fra skreddatabasen.

3.5 Implementasjon av D360-algoritmen

For å teste hypotese tre og sammenligne D3 med en annen type flytalgoritme, ble D360-algoritmen brukt. Programmet ble utvidet med følgende rutinene *slideD360* og *getAspect* (Figur 25), som er basert på kode av Bård Romstad (UiO). Flere av støttefunksjonene som benyttes av D3 brukes også her (Figur 24).



Figur 25: Struktur for D360-algoritmen



Figur 26: Flytdiagram for D360-algoritmen

Algoritmen tar *steglengde*, *maks antall steg* og *maks tillat vinkel* som input. Algoritmen har maks antall steg som avbruddskriterium, men kan også utvides med h/l slik som skissert for D3-algoritmen. *Maks tillat vinkel* begrenser hvor mye hver nye retning kan avvike i forhold til den gamle, på samme måte som hos D3, bortsett fra at her kan begrensningen variere kontinuerlig mellom 0° og 360°. Først blir maks tillatt vinkel konvertert til radianer med formelen:

$$\alpha_{rad} = \pi * \frac{\alpha_{deg}}{180} \quad (\text{Formel 8})$$

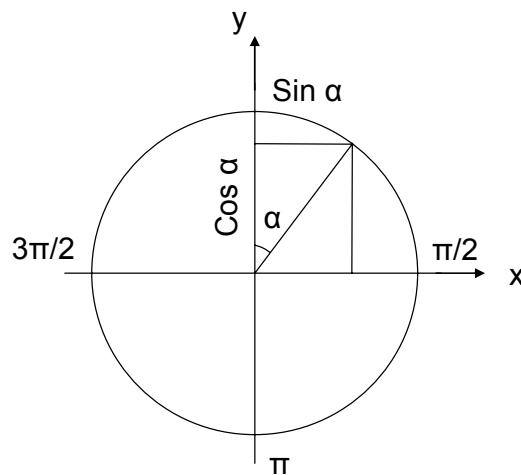
Det bevegelige vinduet fylles med verdier som skissert for 3x3, men i dette tilfellet brukes ikke *Direction*-objektene, kun verdiene i vinduet direkte (Figur 16). Disse verdiene benyttes i *getAspect* til å finne parameterne z_x og z_y som regnes ut slik Zevenbergen og Thorne har skissert det (Burrough og McDonnell 1998, Wilson og Gallant 2000). I denne algoritmen brukes aspektet som en indikator på flytretning og formelen er hentet fra Wilson og Gallant (2000):

$$\Psi_{FD} = 180 - \arctan\left(\frac{z_x}{z_y}\right) + 90\left(\frac{z_x}{|z_x|}\right) \quad (\text{Formel 9})$$

Aspektet gies i radianer med klokka med nord som utgangspunkt (Figur 27). Oppdateringsverdiene i x og y-retningen hentes ut av aspektet med følgende formler:

$$x = \sin(\alpha), y = \cos(\alpha) \quad (\text{Formel 10})$$

I motsetning til det lokal koordinatsystem for D3 (row og col), har koordinatsystem utgangspunkt i nedre venstre hjørne, noe som gjør håndtering enklere. x og y ganges så med steglengden, og koordinaten oppdateres.



Figur 27: Oppdateringsverdiene i x og y-retningen.

3.5.1 Framgangsmåte for sammenligning

D3-algoritmen og D360-algoritmen blir først bli testet mot hverandre på genererte terrengmodeller. Modellene ble lagd i Arcinfo ved hjelp av funksjonen *rasterkalkulator* og

enhetene $colmap$ og $rowmap$, som representerer x og y , henholdsvis. Det blir generert et plan som heller mot nordvest med formelen $f = (x, y)$ (origo er i øvre venstre hjørne), et plan som heller mot nordnordvest med formelen $f = (x+2y)$. Det blir også generert en bolle, med senter for bollen i origo, med formelen $f = (x^2 + y^2)$ og en omvendt bolle med formelen $f = -(x^2 + y^2)$. En spesiell terrengformasjon blir generert basert på punktdata etter mønster fra Tarboton (1997). Dette er for å teste egenskapene til D360-algoritmen på en terrengformasjon som medfører vanskeligheter for flytalgoritmer, slik som 2D-Lea og DEMON.

D3 og D360 blir også sammenlignet på samme område med både ASTER og SKT25 terrengmodellene. Først blir D3 kjørt på SKT25 og siden ASTER og resultatene legges oppå hverandre. Dette blir også gjort for ASTER. Disse terrengmodellene ble prosessert med en fyll-algoritme. Det blir testet for de samme områdene som ble testet i forbindelse med hypotese en. En begrensning her er at ASTER-terrengmodellen ikke er glattet.

4 Resultater

I første del av kapittelet viser hvordan hypotese en testes, ved at D3 sammenlignes med reelle skred. Problemer med algoritmen blir diskutert. Det blir også gjort en sammenligning med D8 for å se om det var noe vinne på implementasjonen av D3. Undersøkelsene gjøres ved åtte skredlokasjoner, samt ved Geirangerfjorden. Videre undersøkes forekomsten av ties i to terrengmodeller. Konsekvensen av ties blir så testet, både for et område nær Geirangerfjorden og de samme åtte skredlokasjonene. Videre blir D3 og D360 sammenlignet, på kunstige flater og reelle terrengmodeller av flere typer, samt for reelle skred. I del fire føres det en diskusjon over konseptene modell og algoritme, og hva det egentlig vil si at en modell er mer velegnet enn en annen.

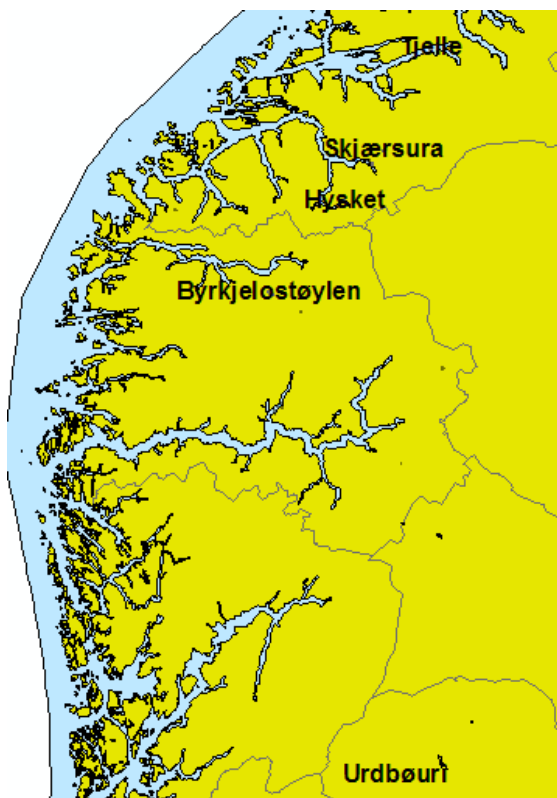
4.1 Sammenligning med virkelige skred

Det ble utført en sammenligning av skreddata fra skreddatabasen med D3-algoritmen og D8-algoritmen som beskrevet i metodekapittelet. Det ble videre utført en sammenligning med skred ved Geirangerfjorden digitalisert kun som linjer. Viktigere er en sammenligning av skred som er digitalisert i hele sitt omfang. Åtte steder, hvor skredene er digitalisert som polygoner, ble valgt ut: Skjærsura, Byrkjostøylen, Hysket, Urdbøuri, Tjelle og Langhammern i Sør-Norge og Gumpedalen og Kjusakdalen i Nord-Norge. Resultatene er presentert i figurene nedenfor (Figur 30 til Figur 37). Disse analysene ble utført med både D3 og D8-algoritmen, men resultatene fra disse to algoritmene var nesten utelukkende sammenfallende, med unntak av noen resultater som blir diskutert senere. I den videre diskusjon refereres det til D3-algoritmen når det menes D3 og D8 slik de er implementert i denne oppgaven, med mindre noen annet er presisert.

Figur 38 viser digitaliseringen av noen skred langs nordvestsiden av Geirangerfjorden. D3 følger disse med stor grad av treffsikkerhet, noe som styrker hypotese en, da simuleringene kan sies å gi anvendbare resultater. Et unntak vises i detalj i Figur 39. Et startpunkt identisk med det digitaliserte skredet gir et avvik på over 200 meter når skredet treffer fjorden. Mens et startpunkt litt bortenfor følger skredbanen svært godt. Dette kan skyldes en unøyaktighet i terrengmodellen, en unøyaktighet i digitaliseringen, eller problemer med algoritmen. (Huggel et al. 2003b) hevder at unøyaktigheter som regel skyldes feil i terrengmodellen, framfor feil i algoritmen.

Startpunktet for algoritmene er valgt manuelt, og forsøkt plassert så sentralt som mulig på løsneområdet. Valg av startpunkt ser ikke ut til å variere resultatene i stor grad, før det nærmer seg sidekantene av løsneområdet. Hvis simuleringen kan sees som et gjennomsnitt av skredet, et sted hvor momentet er størst, er det mest fornuftig å velge et punkt nær sentrum.

Nær kantene av løsneområdet får man ofte et avvikende resultat (se Figur 40). Figuren viser at i dette området er helningen annerledes. Et skred som har midten av sitt løsneområde her, bør antas å ha et annet forløp.

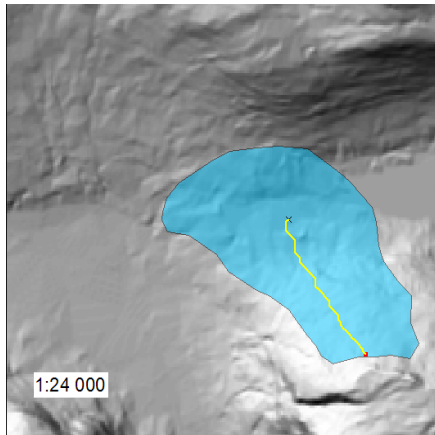


Figur 28: Utvalgte skred i Sør-Norge

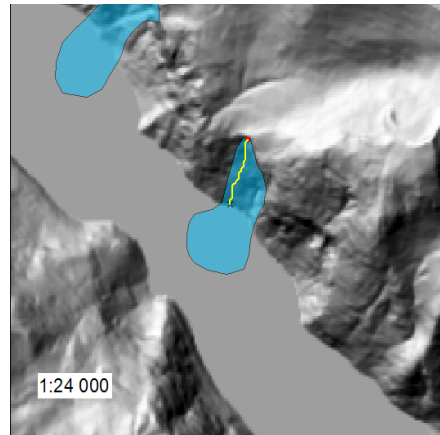


Figur 29: Utvalgte skred i Nord-Norge

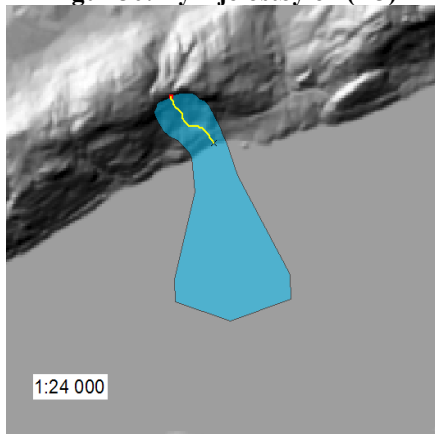
For Langhammern, Tjelle og Hysket (Figur 31, Figur 32 og Figur 34) følger flytsimuleringen i en linje sentrert i skredet og når helt ut til vannkanten, hvor den stopper siden vannet er satt som barriere. De virkelige skredene har stor utstrekning under vann, men simuleringen av dette faller utenfor rammene til oppgaven. I disse tre tilfellene kan D3 sies å gi en brukbar skredsimulering, noe som underbygger hypotese en.



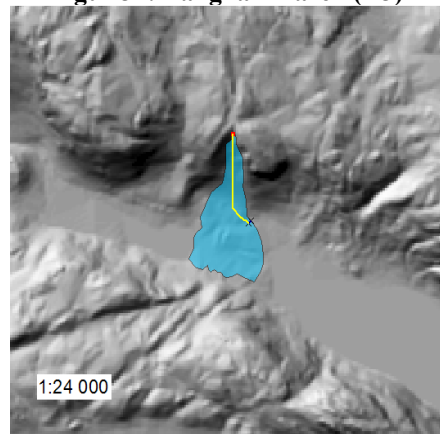
Figur 30: Byrkjeloelstøylen (D3)



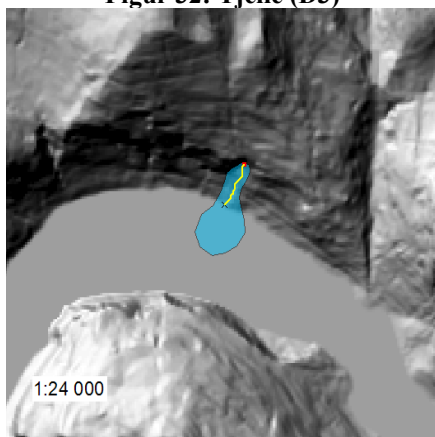
Figur 31: Langhammaren (D3)



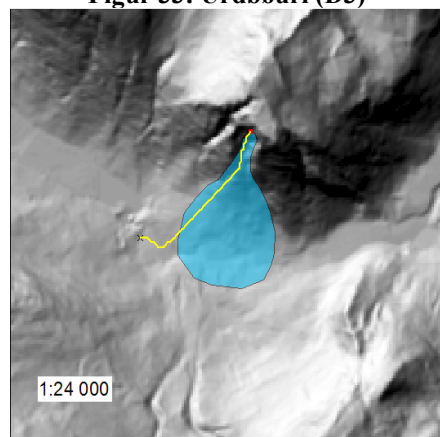
Figur 32: Tjelle (D3)



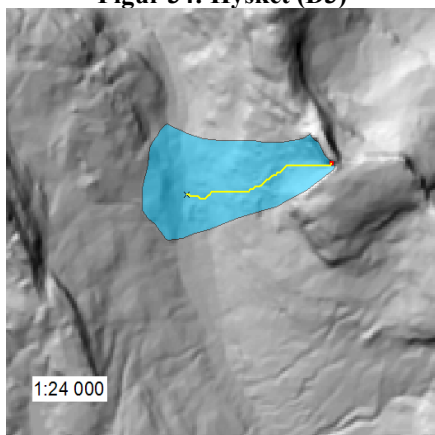
Figur 33: Urdbøuri (D3)



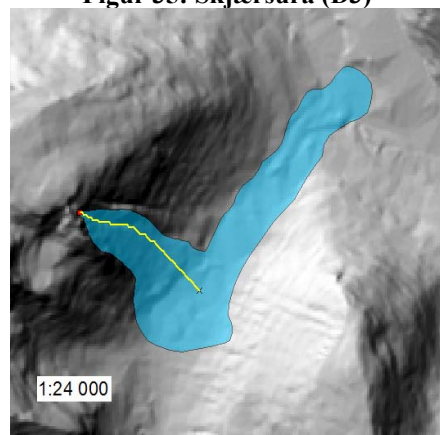
Figur 34: Hysket (D3)



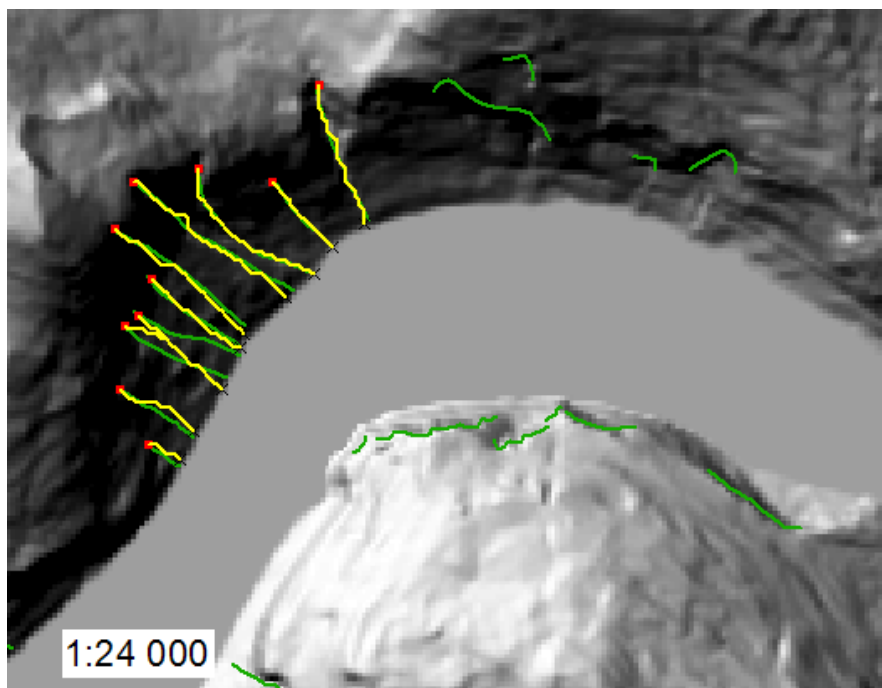
Figur 35: Skjærsura (D3)



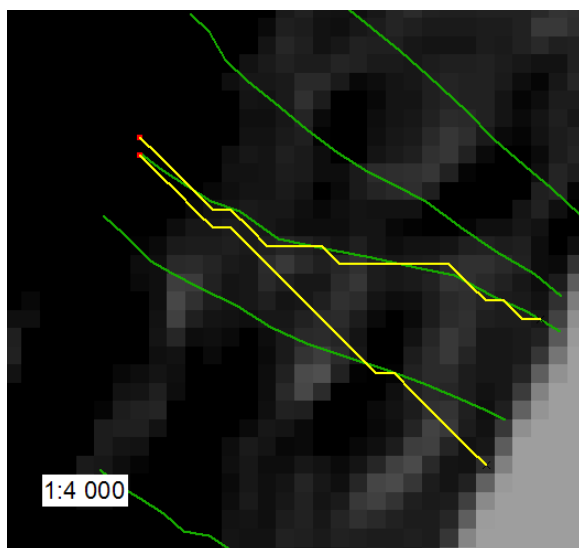
Figur 36: Gumpedalen (D3)



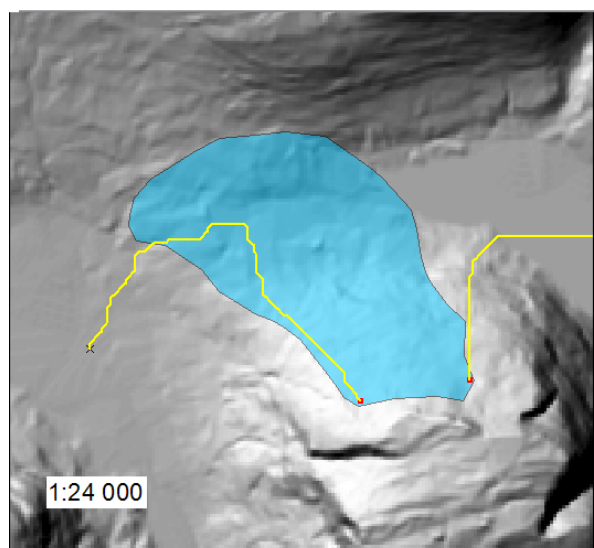
Figur 37: Kjusakdalen (D3)



Figur 38: Digitaliserte skred i Geirangerfjorden (grønn) og D3 (gul).



Figur 39: Digitalisert skred (grønn), D3 (gul).



Figur 40: Simulering fra kanten av løснеområdet.

4.1.1 Skred som går oppover i terrenget

Gumpedalen og Byrkjostøylen (Figur 36 og Figur 30) viser også simulerringer som går rimelig sentrert i skredet. Disse skredene går dog ikke ut i en vannkropp og simuleringen stopper i pits før skredets fulle lengde er simulert. For begge steder skyldes dette at terrenget der skredet ender, heller oppover, og skredet har hatt så mye moment at det har fortsatt oppover dalsiden på andre siden. Dette tar ikke flytalgoritmen høyde for, og all simulering

stopper nederst i dalen. I tilfeller som dette vil man fortsatt få pits i bunnen av dalen, selv om man hadde kjørt en fyll-algoritme på terrengmodellen. Dalen drenerer ut på tvers av skredretningen, og derfor vil en algoritmen som tillater nok avvik til siden og ikke har andre stoppebegrensninger, simulere en sti som fortsetter nedover dalen. Etter å ha kjørt en fyll-algoritme på terrengmodellen, indikerer simuleringen nettopp dette. I Gumpedalen vil simuleringen, på en fylt terrengmodell, tendere til å følge dalen i nordgående retning. Dette kan man tolke som en potensiell skredbane for gjentatte skred. Det kan sies at dette svekker hypotese en, men mangel på mulighet til å simulere flyt oppover er en generell begrensning ved alle eksisterende flytalgoritmer. Dermed er det ikke noe å vinne på å innføre en mer kompleks algoritme. Dette styrker hypotese tre.

Det som kunne supplere algoritmen i et slik tilfelle var at man inkluderte en estimering av hastighet for hvert punkt (se teorikapittelet). Selv om simuleringen indikerer at skredet stopper nederst i dalen, fordi man støter på en pit, vil hastigheten gitt i punktet indikere at skredet ikke stopper her. I de fleste sammenhenger er det dog viktigst å kartlegge om skredet kommer ned i dalbunnen eller ikke. Det er her infrastruktur som oftest vil begynne seg, og eventuelle elveløp som kunne blokkeres av skredet og utløse en ny katastrofe (Huggel et al. 2004). Simuleringen ved Urdbøuri (Figur 33) viser også en sentret linje, men som stopper for tidlig. Heller ikke her hjelper det å fylle igjen pits, for man har begynt å komme ut på flatt terreng og algoritmen tillater ikke en dreining ved lik høyde.

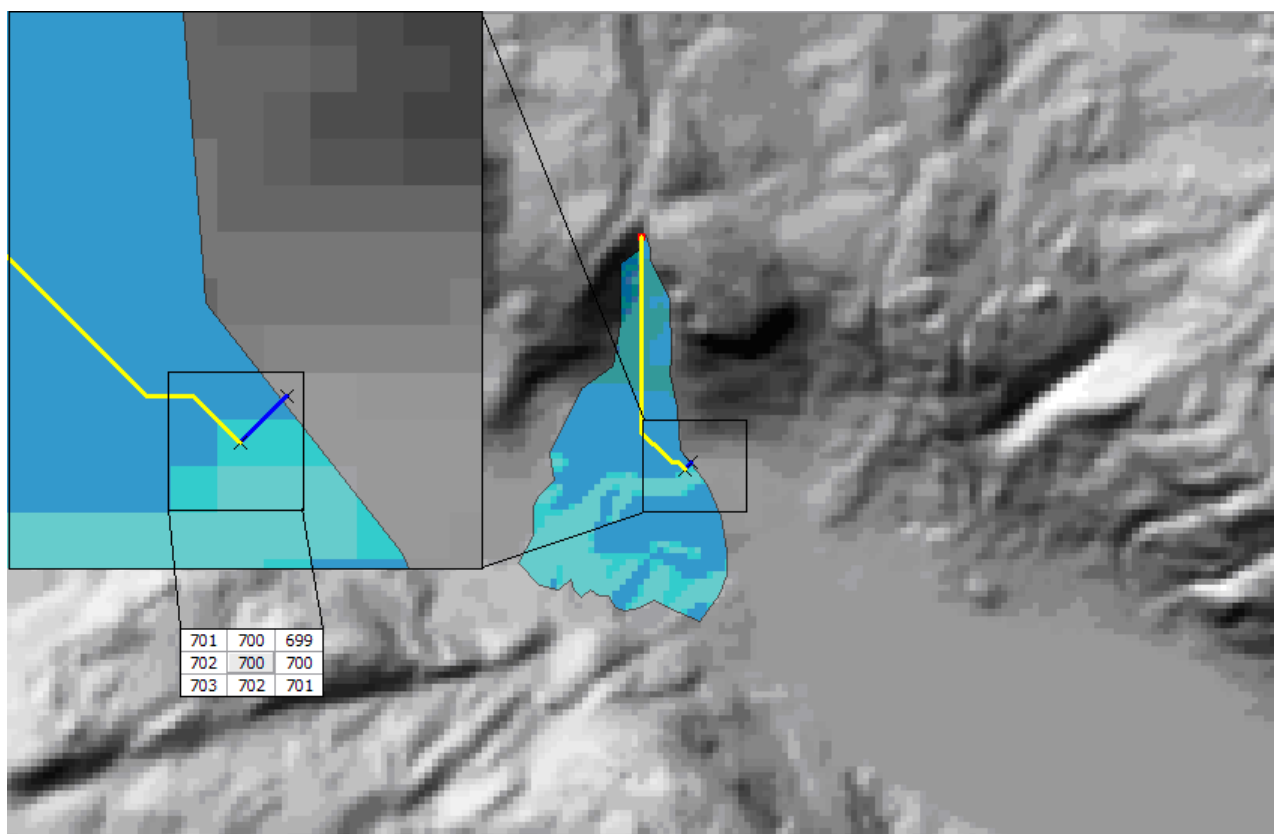
4.1.2 Begrensning ved lik høyde

Kjusakdalen viser derimot et skred som har fortsatt nedover dalen på tvers av den originale skredretningen. Her stopper også skredet midt i dalen, selv etter at alle pits er fylt igjen. Man ville forvente at bevegelsen nedover i dalen ville fortsette, så lenge ikke andre avbruddskriterier stopper banen. Grunnen til dette er begrensningen som ble valgt for tillat avvik (se metode kapittelet). For D3-algoritmen tillates det bare avvik på maks 45°, mens D8 tillater avvik i alle retninger. Men for begge tillates det bare å bevege seg forover hvis høyden er lik som forrige celle. Dette var gjort med argument i skredets moment. Men fyll-algoritmen fyller pits opp til lik høyde med første nabo. Det betyr at mulige utveier for algoritmene kan finnes ved lik høyde, men bare hvis de er *rett fram* i forhold til forrige retning. I dette tilfellet synes det som en klar begrensning. Hvis man i en regional analyse skal ta høyde for verst

tenkelig scenario, så tyder resultatet fra Kjusakdalen på at man utelukker viktige resultater ved å kunne tillate bevegelse over like høyder i samme retning som forrige retning.

4.1.3 Forskjell på D3 og D8

Det viser seg at i de aller fleste tilfeller følger D3 og D8, slik de er implementert i denne oppgaven, identiske løp. Den eneste forskjellen er begrensning i tillatte retninger. Et unntak vises i Figur 41. D3 sjekker ikke den mulige utveien fordi den ligger 90° til siden for forrige retning. D8-algoritmen finner denne utveien, men til liten hjelp, siden det straks dukker opp en pit. Denne forskjellen viser et eksempel på hvordan D3 og D8 kan avvike. Dette er en situasjon som dukker opp sjelden (det eneste tilfellet i de viste eksemplene). Hvis man ser bort ifra naboceller med lik høyde (se forrige avsnitt), så får valget mellom D3 og D8 trivielle konsekvenser.

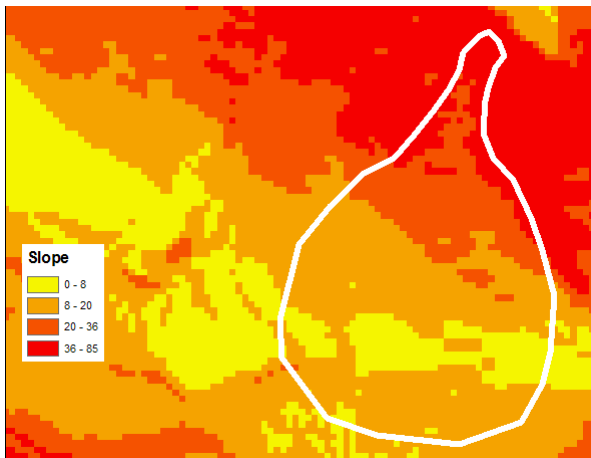


Figur 41: D3 vises som gul linje og D8 som blå. I firkanten vises høydeverdiene til nabocellene.

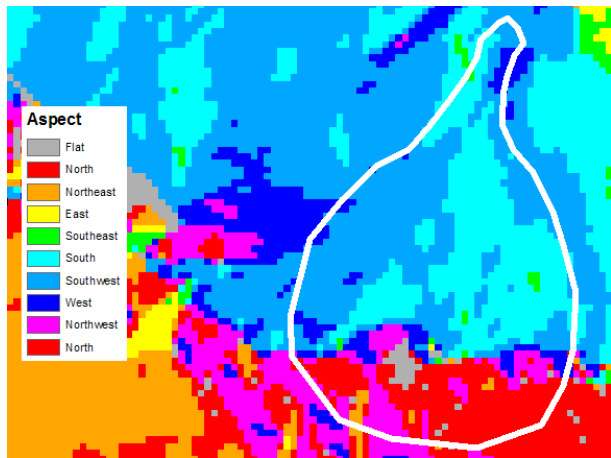
4.1.4 Grid-effekt

For skredet i Skjærsura (Figur 35) viser simuleringen avvikende oppførsel. Dette er et skred som begynner i bratt terreng og fortsetter ut på en konveks vifte (Figur 42). Det er på slikt terreng man ofte får problemer med D8 lignende algoritmer (Huggel et al. 2003b). Banen viser et klart avvik i vestlig retning. Denne tendensen holder stikk, selv med variasjon i valg av startpunktet. Det indikerer også en utbredelse i denne retningen, som det reelle skredet ikke fulgte. Figur 43 viser at aspektet på vestsiden av raset heller mot vest, og det er dermed klart at en sti som beveger seg inn i dette området, vil tendere til å avvike mot vest.

Dette viser en av de klare begrensningene med D8-lignende algoritmer, retningene er diskretisert i åtte utvalgte retninger med 45° mellomrom. Men en annen algoritme (Figur 74), viser faktisk likefallende resultater som D3 her, noe som styrker hypotese tre. Aspektet, slik det er vist i Figur 43, viser en brå overgang mellom sør og sørøstlig aspekt ved enden av ”halsen” til skredet. Det kan tenkes at man ikke har simulert aspektet vellykket i forhold til virkeligheten, eller at lokale forhold har hatt stor påvirkningskraft ovenfor resultatet.



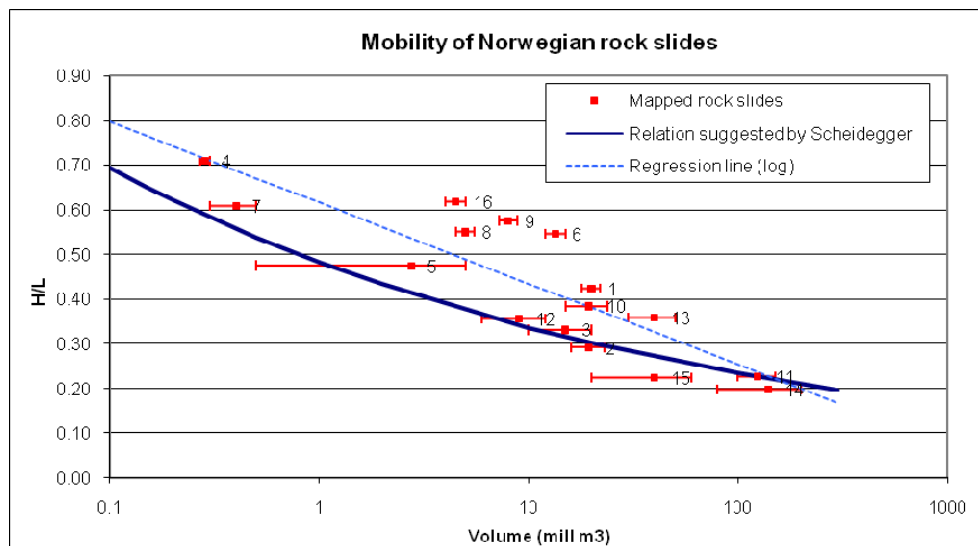
Figur 42: Helningsgrad (Slope)



Figur 43: Helningsretning (Aspect)

4.1.5 Avbrudd

For simuleringene ovenfor var et avbruddskriterium $h/l = 0,02$ satt (se metode kapittel). Sammenfallende for alle simuleringene var at dette kriteriet aldri slo til. Alle skredene begynte i svært bratt terreng, og fortsatt til de møtte en pit eller barriere. Flere av skredene hadde blitt simulert med samme resultat, selv med en $h/l = 0,05$. 0,02 er et veldig romslig kriterium. Figur 44 viser hvordan dette tar høyde for selv de mest ekstreme skredvolum.



Figur 44: h/l og skredvolum i Norge (Romstad et al. 2006).

De skredene som stoppet for pits, kan man skille mellom to typer; de som stopper for naturlige pits og de som stopper for feil i terrengmodellen. Sistnevnte gruppe ønsker man å unngå. Dette kan man unngå ved å kjøre en fyll-algoritme og tillate avvik i alle retninger, også ved samme høyde. Men dette vil fortsatt ikke være nok til å takle alle tilfeller, slik som i Kjusakdalen, hvor skredet har fortsatt nedover dalen,. I slike områder kan man få oppfylte pits som har blitt til flate områder, men simuleringen kan likevel lage en bane som fortsetter rett frem over et slik område, og ikke til siden, slik man kunne ønske. Dette viser at skred kan ha svært kompleks oppførsel, og at det kan være vanskelig å ta høyde for alle situasjoner.

4.2 Ties

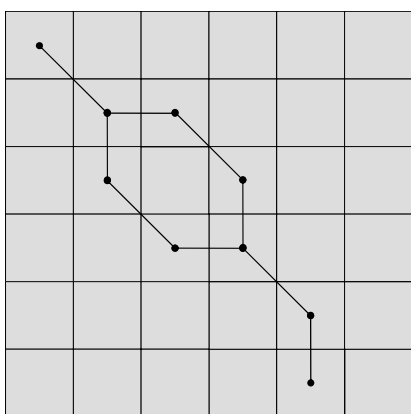
En singelflyt algoritme velger kun en retning. Et betimelig spørsmål er hva som skjer hvis flere enn en celle oppfyller kriteriet. I et grid med cellestørrelse på 25 meter og som dekker hele Geirangerfjorden vil for eksempel ha 533×847 raster = 451 451 celler. Hvis maks høyde er for eksempel 1200 m i rasteret, betyr det at vi har et utfallsrom på $[0, 1200]$. Hvis man antar alle cellene som terrengceller (ikke noe vann) og at terrenghøydene hadde fordelt seg jevnt innenfor utfallsrommet, hadde det gitt ca 376 celler for hver enkelt høyde. Når vi forutsetter at topografien varierer jevnt, er sannsynligheten rimelig for enkelte like utfall dukker opp. Siden det brukes distansevektning, deles høydeforskjellene i de diagonale retningene på roten av to, og mange desimaler introduseres. Skal resultatet være matematisk korrekt, så bør det være like mange gjeldende siffer i vektingen som i høydeverdien. Ellers får man en situasjon, hvor man fjerner muligheten for at det blir like verdier i diagonale retninger

(dette blir statistisk umulig med 14 desimaler hvis roten av to lagres som en variable med formatet double (Microsoft 2008), mens man beholder den potensielle muligheten i de kardinale retningene.

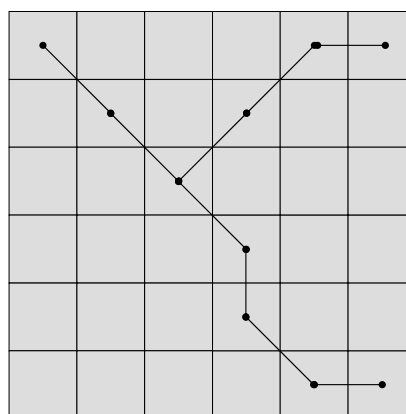
Er det to eller flere celler som oppfyller kriteriene for bratteste skråning, sier vi at resultatet er uavgjort, eller en tie. Hvis det finnes to eller flere slike celler, vil det for en singelflytalgoritme være den første retningen som blir valgt. Hvis ikke noe annet blir gjort, vil algoritmen overse potensielle stier. Dette vil ikke skje på en tilfeldig måte, og dermed potensielt gi en tendens til at stier går mot høyre i nordvendte skråninger hvis algoritmen startet med øst og sjekket retningene mot klokka.

Hvis slike forekomsten av slike ties er forsvinnende liten, kan dette forsvares. Tarboton forutsetter følgende: *“In nature ties are extremely rare so the bias introduced by this is deemed negligible”* (Tarboton 1997). Men Goodchild hevder at *“DEM data contains large number of ties because vertical resolution is not high”* (Goodchild 2008).

Det neste poenget er om konsekvensen av slike ties er stor eller liten. Figur 45 og Figur 46, viser hvordan konsekvensen av ties hypotetisk enten kan føre til enten konvergerende eller divergerende oppførsel.



Figur 45: Konvergerende oppførsel

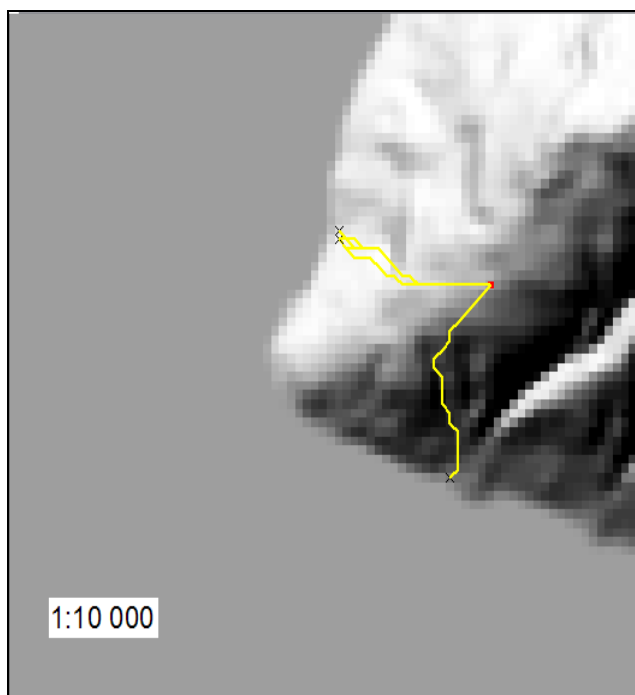


Figur 46: Divergerende oppførsel

Man kan tenke seg at konvergering ofte vil bli resultatet i skråninger, spesielt med klare avgrensninger på hver side. Men på konvekse flater kan divergerende stier ofte bli resultatet. Konsekvensen av ties beror på to ting, hyppighet og konsekvens. Hvis en av disse faktorene er neglisjerbar, kan det hevdes at det ikke er bryet verdt å utforske ties. Men selv et lite antall

ties, hvis det viser seg at de potensielt har store konsekvenser, bør taes med i en analyse hvis man ikke ønsker å miste noen resultater.

Figur 47 er en simulering, å ikke basert på noe kjent løsneområde, men viser hva en tie potensielt kan føre til. Skredbanene får totalt forskjellige løp, og treffer på hver sin side av et nes, noe som også kan få konsekvenser for en bølgesimulering (Harbitz 2002).



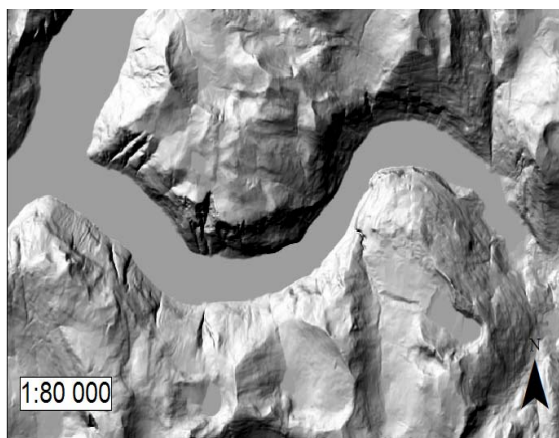
Figur 47: Dramatisk følge av divergerende skredbane

4.2.1 Terskelverdier

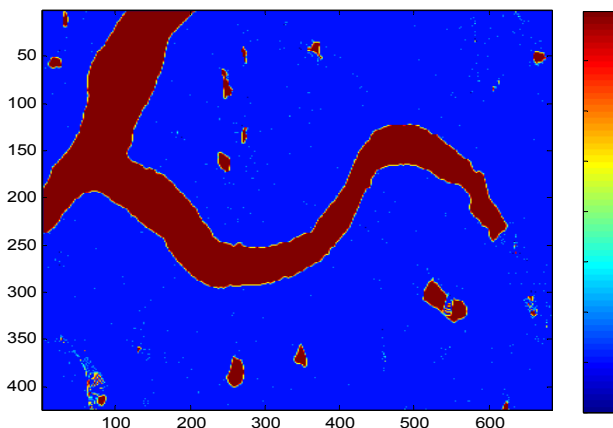
En terrengmodell for Geirangerfjorden basert på SKT25 (Figur 48) ble undersøkt for mulige ties (Figur 49). De store røde områdene er fjorden eller mindre innsjøer. I slike flate områder vil man få ties i alle retninger. I grenseområdene vil det også være mange ties, vist som gule kanter langs vannkroppene. Disse er ikke interessante. Men i områdene mellom er det enkelte forekomster. Det kan sies å være et ikke trivielt antall.

Definisjonen av ties gitt ovenfor er basert på en eksakt likhet. Men Tarboton har naturligvis rett i sin påstand om at ties forekommer sjelden i naturen. Det er bare et spørsmål om målenøyaktighet. Presisjonen til høydeverdiene i en terrengmodell er ikke eksakt. Hvis man tar dette perspektivet, er det ikke urimelig å godta alt innenfor et vist avvik som ties. Hva

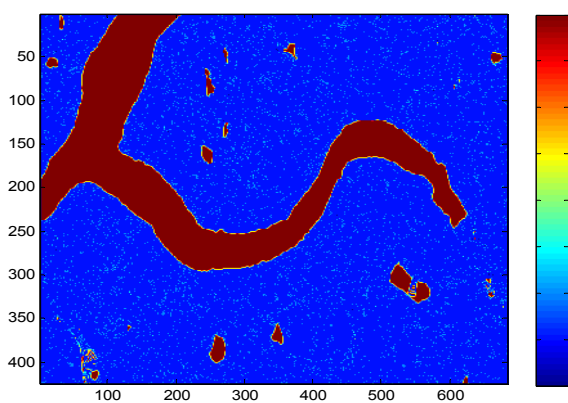
grensen for dette avviket er kan sees på som en terskelverdi. Det er to måter å sette denne terskelverdien; en absolutt verdi og en relativ verdi avhengig av skråningen. En relativ verdi kan for eksempel være 1% av høydeforskjellen delt på distansevekten (altså skråningen). Hvis terskelverdien er relativ, vil antall ties variere konstant utover terrengmodellen. Hvis derimot terskelverdien er absolutt, satt til for eksempel 0,01, vil ties forekomme mye hyppigere på flate områder. Figur 50 og Figur 51 viser antall ties i Geirangerfjorden, denne gangen med terskelverdier lik henholdsvis 0,02 og 0,05. Disse verdiene er relative. En verdi på 0,02 medfører at alt innenfor 2% regnes som tie. En skråning på 0,5 (45°) medfører at verdier mellom 0,5 og 0,49 vil regnes som tie. De blir gradvis mer tallrike når terskelverdien øker, og ser ut til å være spredt tilfeldig. Men det ser ut til å være noen områder (blå felter), hvor det fortsatt nesten ikke forekommer ties.



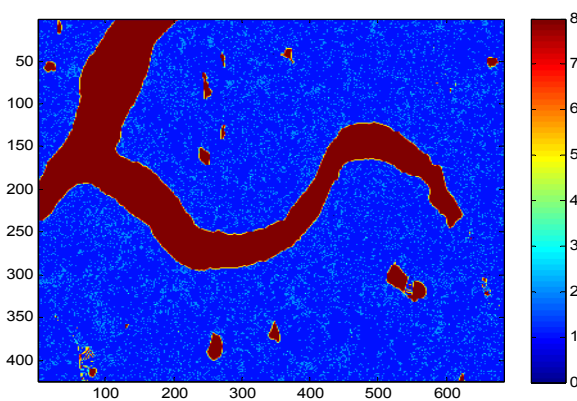
Figur 48: Geirangerfjorden.



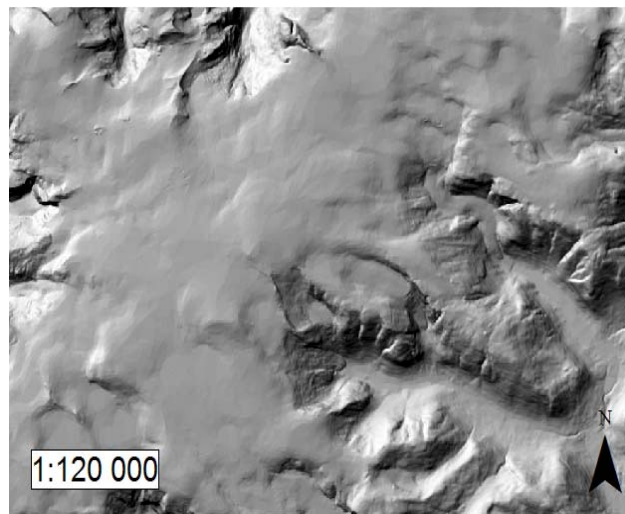
Figur 49: Terskel = 0.



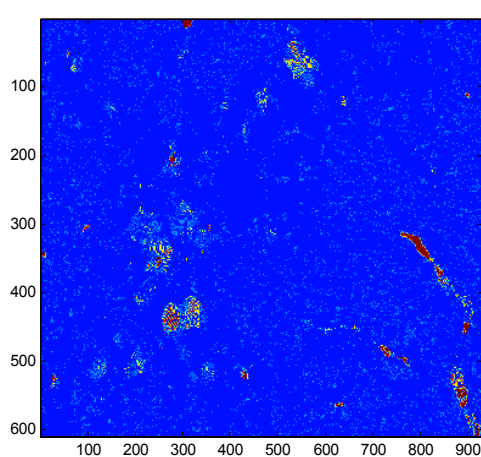
Figur 50: Terskel = 0,02.



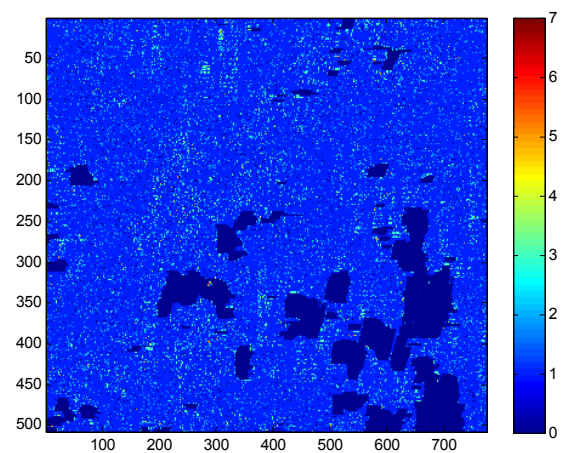
Figur 51: Terskel = 0,05.



Figur 52: Jostedalsbreen



Figur 53: SKT



Figur 54: ASTER

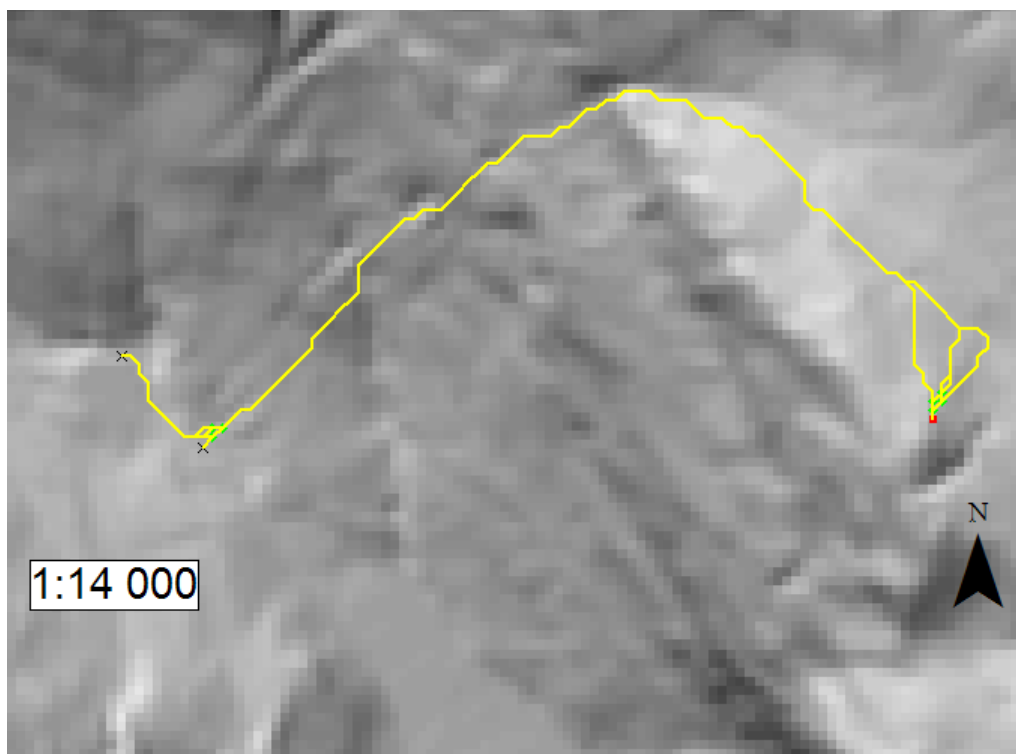
Ytterligere et område ble undersøkt. Figur 52 viser et utsnitt av den nordlige delen av Jostedalsbreen. Antall ties ble undersøkt, både for SKT25 og en ASTER-generert terrengmodell (Figur 53 og Figur 54). Terskelverdien her er satt til 0,02. De store mørkeblå områdene i ASTER-terrengmodellen skyldes manglende data. Røde områder er igjen vann eller sletter (det tydelige røde område midt på høyre del, er Nigardsbrevannet). Igjen kan man se en del områder uten ties. Disse ser ikke ut til å være korrelert med skråning.

Det er tydelig at ties forekommer ofte i en terrengmodell, selv med en terskelverdi på 0,02. Dette styrker i en viss grad hypotese to sitt første ledd; at forekomsten av ties kan være ikke-triviell i gridbaserte terrengmodeller. Det er også interessant å se at det er svært mange i

ASTER-terrengmodellen. Det viser seg å være mye lokal variasjon, og at modellen med fordel burde vært glattet før analysen. Dette støttes av at ASTER-modellen mangler de flate, røde områdene, har mange flere ties og de ser ut til å være mer spredt.

4.2.2 Konsekvensen av ties

Resultatene ovenfor har avdekket at ties forekomsten av ties er signifikant. Hvis konsekvensen av ties viser seg å være at de konvergerer, er konsekvensen av dem neglisjerbar. I første omgang er et område i nederste venstre hjørne av terrengmodellen fra Geiranger undersøkt nærmere med D3R, fordi Matlab-analysen viste at det forekom flere tiepunkter her (Figur 55). Det bør minnes om at utløpslengden her ikke er stoppet av en h/l avbruddsmekanisme. I denne sammenhengen er ikke skredlengden interessant, men oppførselen til banen. Her er terskelverdien satt til 0,02, og er en absoluttverdi.

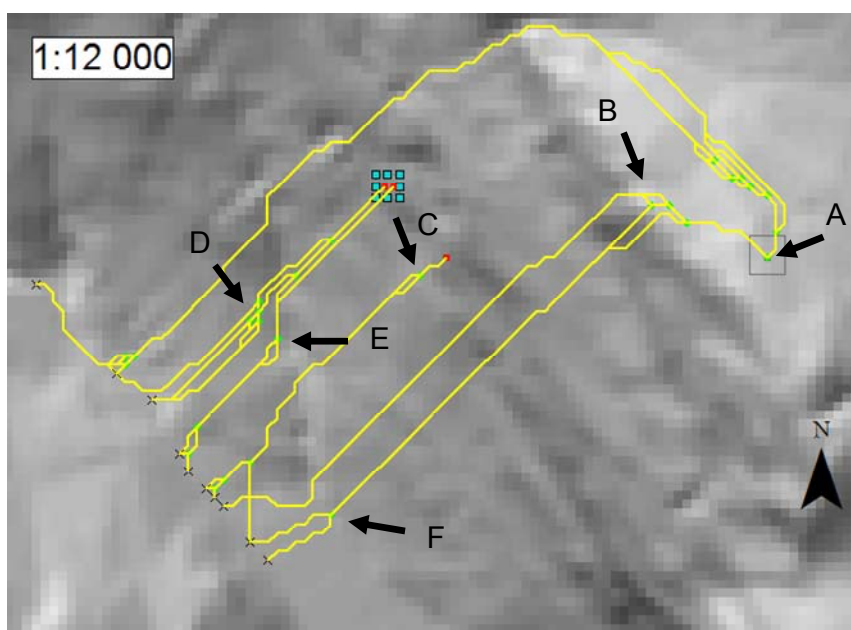


Figur 55: Konsekvens av ties

Grønne punkter er tie-verdier. Her kan man se en tydelig spredning i starten. Men banene konvergerer til en bane. Det er noe spredning mot slutten.

I neste figur viser de fleste pilene gode eksempler på divergerende oppførsel (Figur 56). A er spesielt tydelig, men dette er et toppunkt, og er derfor i en særstilling, siden vi vil forvente divergerende oppførsel fra toppunkt. Dette er også av mindre konsekvens siden løseområder

ikke forventes å være toppunkter. Likevel forekommer det mest konvergerende ties, slik som E og C. Legg også merke til de typiske parallelle banene fra noen ties, som B og F. Eksemplet ovenfor viser at ties forekommer relativt ofte, og at de kan ha både konvergerende og divergerende stier. Andre ledd i hypotese to, som postulerer at konsekvensen av ties sjelden får effekt for resultatet av D3-algoritmen, blir dermed vanskelig å bekrefte eller avkrefte. Likevel gis det en viss støtte for hypotesen, da det i størst grad forekommer konvergerende ties.

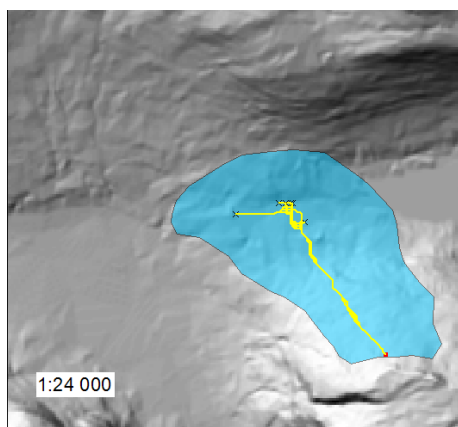


Figur 56: Konsekvens av ties

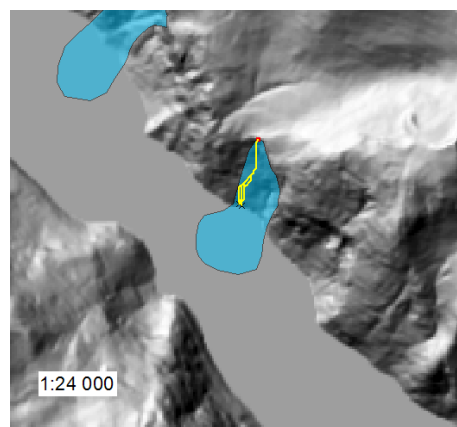
4.2.3 Konsekvensen av ties i virkelige skred

Konsekvensen av ties ble undersøkt for de utvalgte digitaliserte skredområdene (Figur 57 - Figur 64). For Langhammaren, Tjelle og Hysket viser en simulering at ved ties konvergerer banene, og resultatet er nært sammenfallende uansett hvilke retninger man tar. Disse simuleringene svekker også hypotese to. Byrkjelostøylen, Gumpedalen og Kjusakdalen viser en annen trend. Ingen av disse går som nevnt ut i en fjord og skredet går på tvers av en dal, som har lengderetning normalt på skredet. I disse tilfellene kan man se at banen indikerer en videre bevegelse i denne retning. Selv om det bare er Kjusakdalen som har en slik bevegelse, er dette et skredforløp man bør ta høyde for.

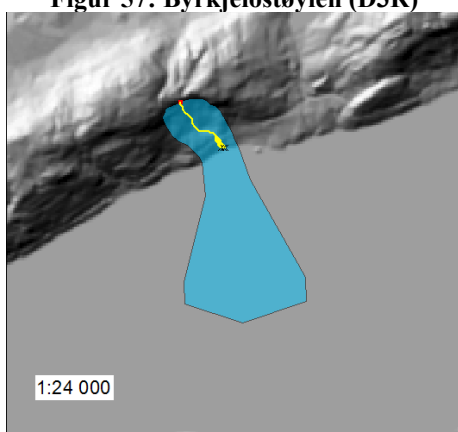
Ved Skjærsura kan man se en klart avvikende bane fra resten, som konvergerer sammen. Kanskje kan den i dette tilfelle indikere en bane som ikke er realistisk, men det viser at ved å



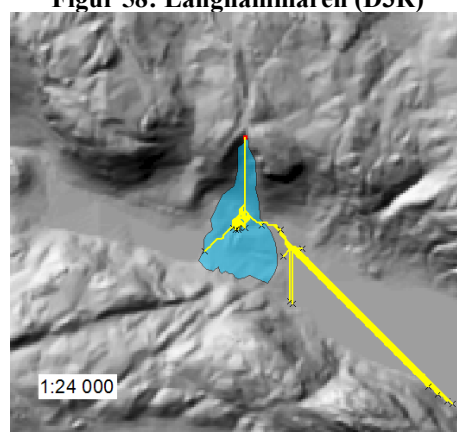
Figur 57: Byrkjeloøstøylen (D3R)



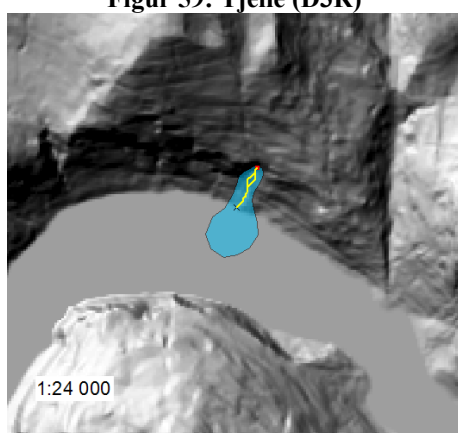
Figur 58: Langhammaren (D3R)



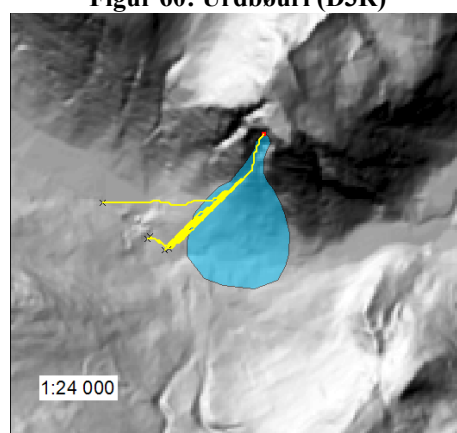
Figur 59: Tjelle (D3R)



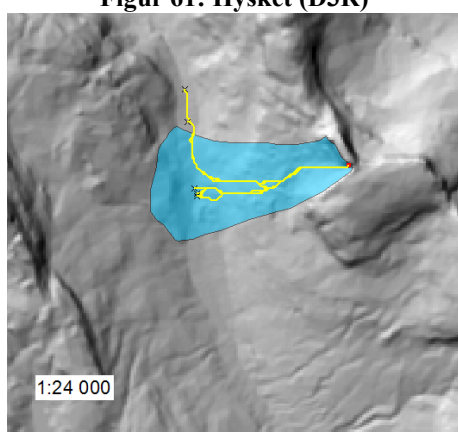
Figur 60: Urdbøuri (D3R)



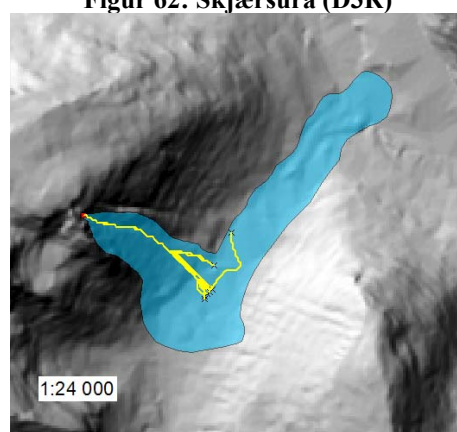
Figur 61: Hysket (D3R)



Figur 62: Skjærsura (D3R)



Figur 63: Gumpedalen (D3R)

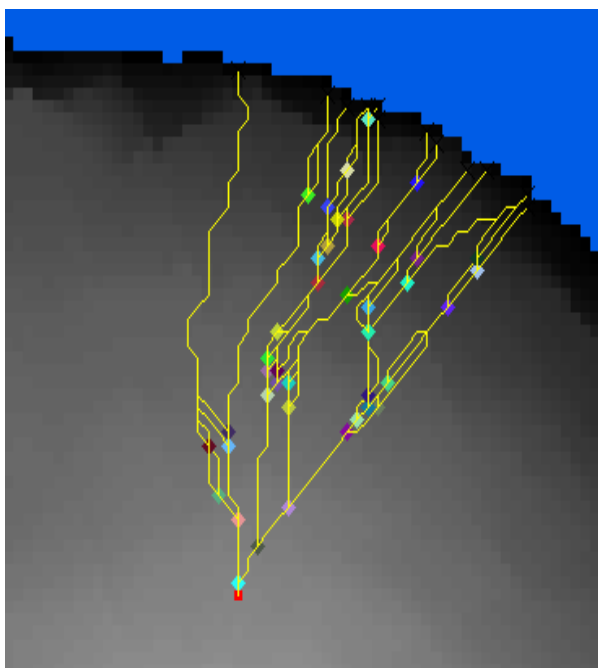


Figur 64: Kjusakdalen (D3R)

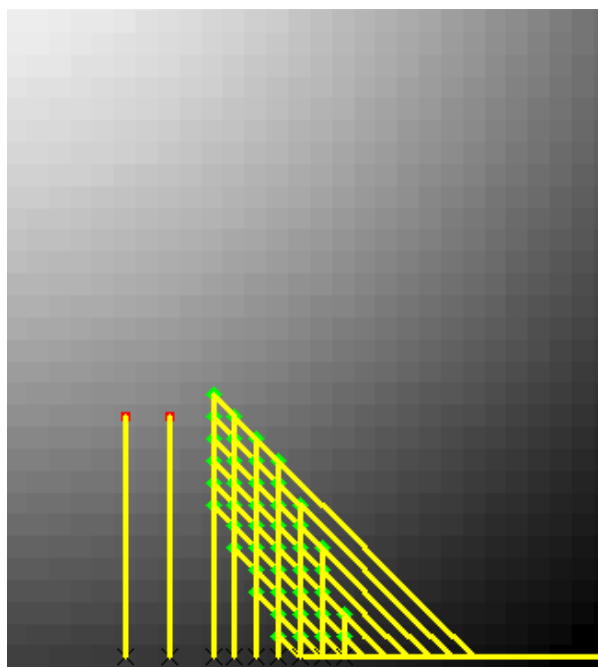
ignorere ties, kan man potensielt miste viktig informasjon. Det mest interessante tilfellet er Urdbøuri. Dette tilfellet avdekker mye spredning rundt et problemområde midt på vifta, og har flere lange utløpere nedover dalen. Her blir konsekvensen av ties divergerende baner. En sammenligning med D3-analysen viser at mye informasjon ville gått tapt her, hvis ikke alle tie-punktene hadde blitt utforsket.

4.2.4 Simulering av spredning

En bi-effekt av undersøkelsen på ties er at med en høy terskelverdi får man en tilnærming til multipleflyt-algoritmer. FD8-algoritmen allokere flyt proporsjonalt til alle lavereliggende celler (Quinn et al. 1991). Denne er blitt kritisert for å gi alt for mye spredning (Tarboton 1997, Huggel et al. 2003b). Hvis terskelverdien settes så høyt at alle lavereliggende celler undersøkes, så er i praksis det samme oppnådd. Men med terskelverdien har man en mulighet til å regulere. Settes denne dessuten som en absoluttverdi, får man den bi-effekten at det blir flere ties i flate områder. Dette er en nyttig egenskap, for D3-algoritmen, i likhet med D8, gjør det dårlig på flate områder.



Figur 65: Tilnærming til MF



Figur 66: Spredning på divergerende overflate

Figur 69 viser hvordan man med en relativt høy terskelverdi (0,5) får simulert spredning i et flatt område. Dette gir en tilnærming på den potensielle utbredelsen til skredet. Algoritmen

D3R preges fortsatt av negative egenskaper ved D3. Figur 66 viser simulering på en kule. Settes terskelverdien for lavt, vil algoritmen tendere til å simulere flyt utelukkende i retninger delelig med 45° . Settes den for høyt, vil det gi for stor spredning.

4.3 Sammenligning av flytalgoritmer

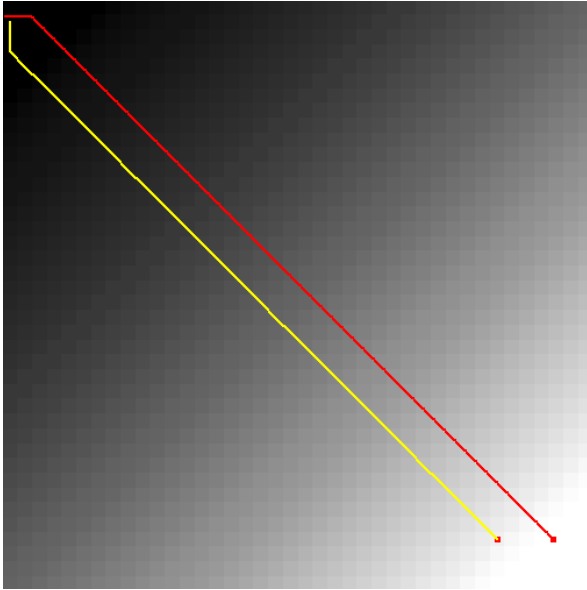
Første del av resultat kapitlet har vist at det er trivielle forskjeller mellom D3 og D8, i alle fall under forutsetningen at begge håndterer naboceller med lik høyde likt. Det finnes flere alternativer til D8-algoritmen. Mange av dem er oppsummert i teorikapitlet.

Likevel er D8-algoritmen eller varianter mest utbredt innenfor hydrologien (Tarboton 1997, 2003, ESRI 2007) og skredforskning (Meissl 1998, Huggel et al. 2004, Romstad et al. 2006). Resultatene skissert ovenfor har vist flere begrensninger ved D3 og D8, som begrensningen til åtte hovedretninger. En algoritme er implementert for å teste om det å la mulige retninger variere kontinuerlig i alle hovedretningene kan gi en forbedring. Denne algoritmen er skissert i metodekapitlet, og er kalt D360. I denne delen sammenlignes D360 med D3 for å teste den tredje hypotesen. Algoritmene sammenlignes først på plane flater og kuler og så på virkelige terrengmodeller, SKT25 og ASTER. De blir også sammenlignet med de digitaliserte skredene, slik det ble gjort i første del av kapitlet.

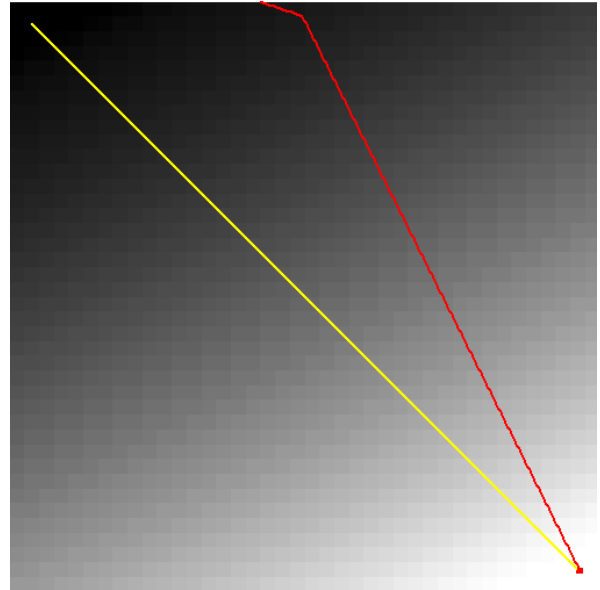
4.3.1 Kunstige overflater

Først ble algoritmene sammenlignet på en plan flate som heller mot NW (Figur 67). Begge algoritmene følger her helningsretningen. Grunnen til at D3 også gjør det, er at planet her heller eksakt i en retning som er delelig med 45° (0° , 45° , 90° , 135° osv.). Hvis helningen ikke er delelig med 45° , forholder det seg annerledes. I Figur 68 heller planet tilnærmet mot NNW³. D360 følger helningen eksakt, mens D3 velger den nærmeste retningen som er delelig med 45° . Dette viser at man kan få et avvik på maks $45^\circ/2 = 22,5^\circ$ i forhold til korrekt retning. Bla. Fairfield og Leymarie har gjort oppmerksom på dette (1991).

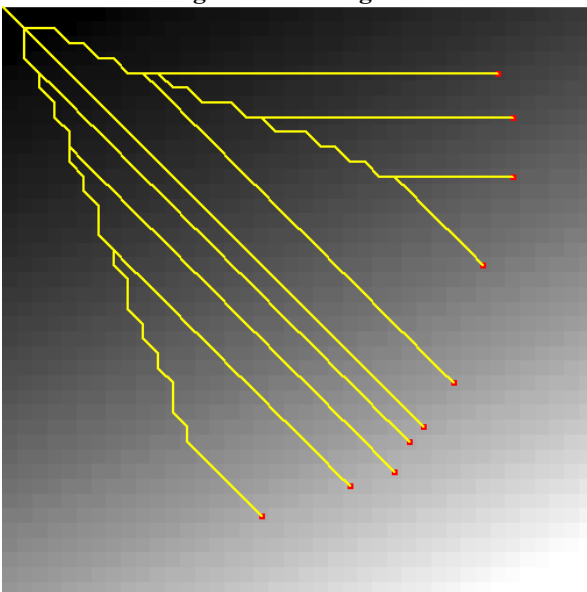
³ Planet er konstruert ved formelen $F(x, y) = x + 2y$. Dette gir ikke en eksakt helning mot NNW, som ville vært 337° , men gir en helning med aspekt på $333,4349^\circ$. Poenget blir dog det samme.



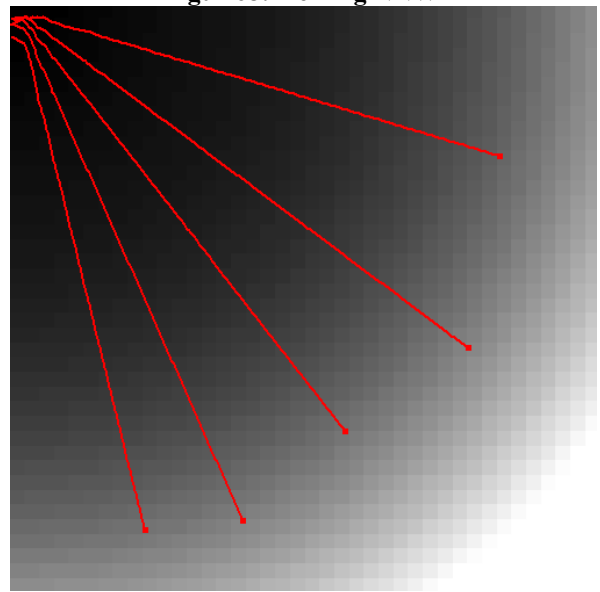
Figur 67: Helning NW



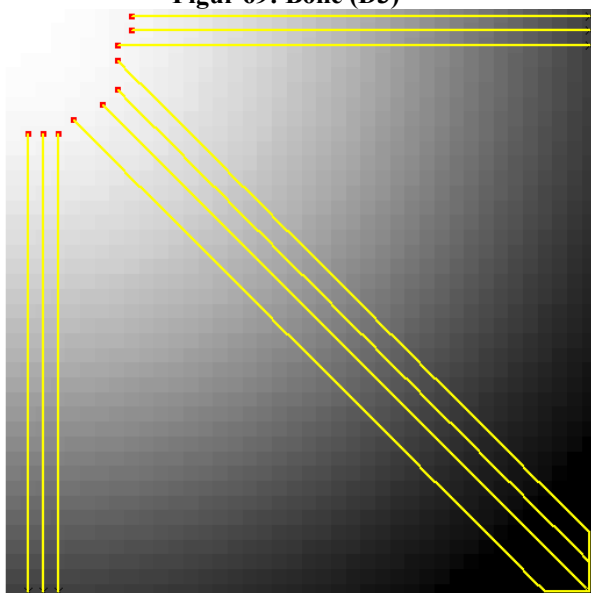
Figur 68: Helning NNW



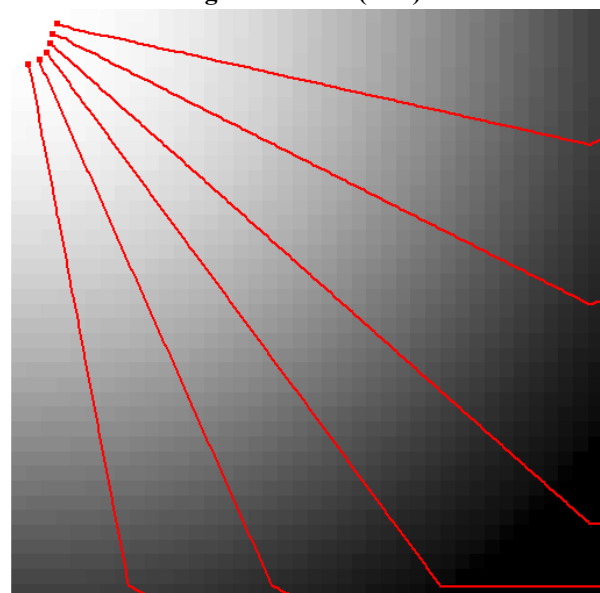
Figur 69: Bolle (D3)



Figur 70: Bolle (D30)



Figur 71: Kule (D3)



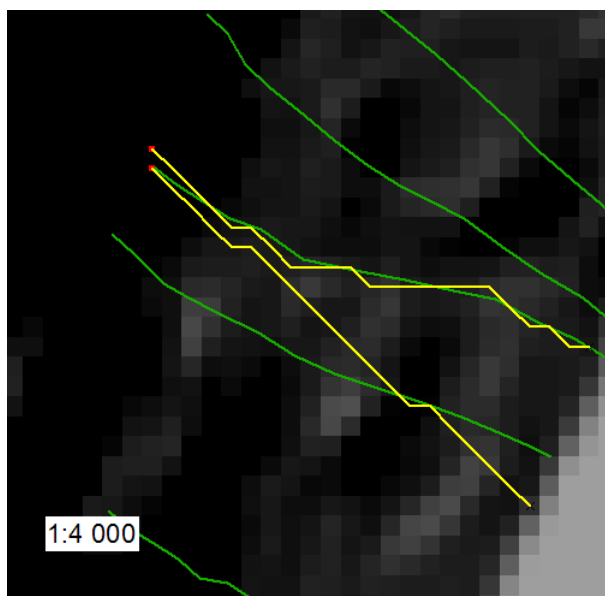
Figur 72: Kule (D30)

I en terrengmodell formet som en bolle, bør alle stier fra kanten av bollen konvergere inn mot midten. I Figur 69 og Figur 70, er øvre venstre hjørne sentrum i en bolle. Man kan tydelig se

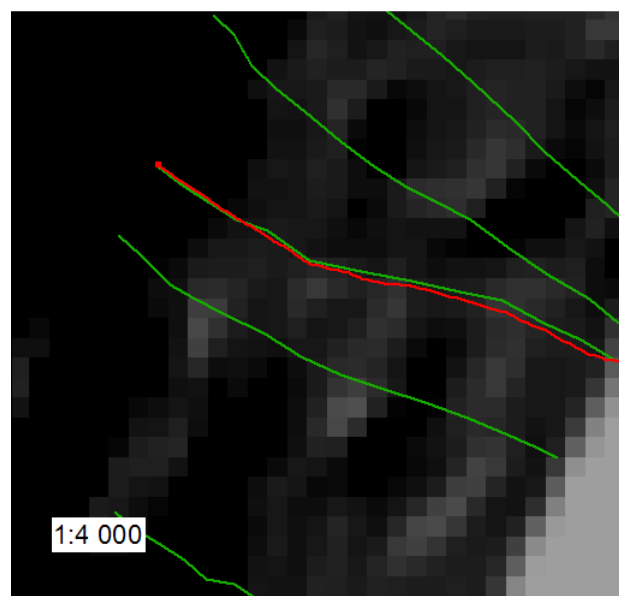
D360 konvergerer inn mot sentrum. D3 konvergerer også, men viser helt klart parallelle linjer underveis, baner som burde konvergere (Wilson og Gallant 2000). Man kan også se tydelig sikksakk-effekt. Men resultatet blir omtrent det samme, tendensen er konvergering mot midten.

Figur 71 og Figur 72 viser en bolle som er snudd på hodet. Igjen er sentrum i øverste høyre hjørne. Her kan man se at D360 velger jevne og korrekte linjer langs overflaten. Men her mener mange at på en konveks flate som denne, er korrekt oppførsel at man skal få spredning. Både D8 og D360 er enkelflytalgoritmer og således ute av stand til å simulere spredning. Her viser igjen D3 en tendens til å skape parallelle linjer langs de diagonale og kardinale hovedlinjene.

Disse simuleringene tyder på at D360 er mye mer velegnet enn D3 og støttes også av flere lignende resultater (Tarboton 1997, Wilson et al. 2007). D360 takler flater som heller i alle retninger, samt både konvekse og konkave overflater, hvis man ser bort fra mangelen på spredning. Dette svekker hypotese 3 om at mer komplekse algoritmer tilfører lite til regionale analyser i forhold til D3 algoritmen. Dog kan det innvendes at slike flater ikke finnes i naturen.



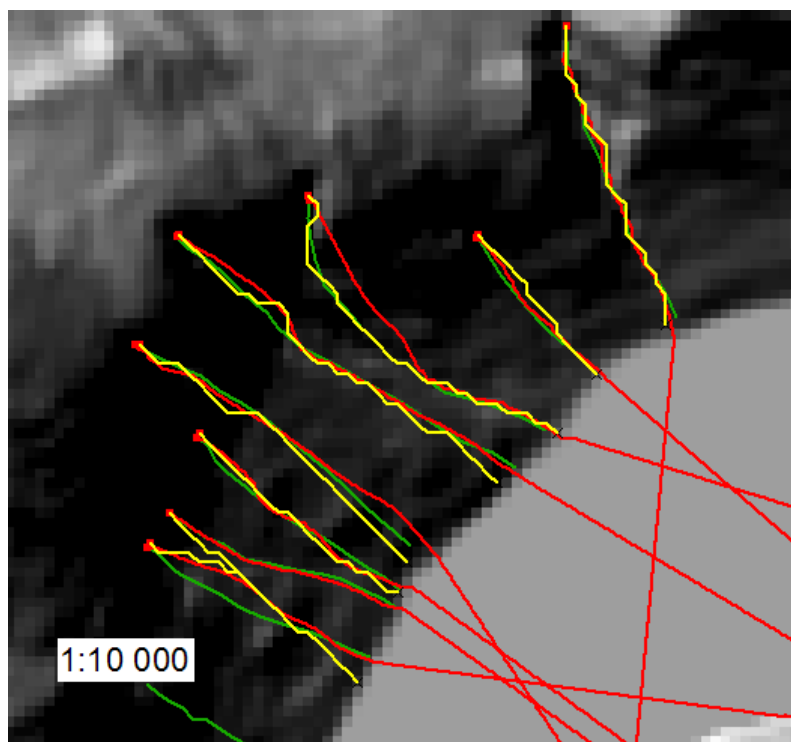
Figur 73: D3- Geiranger



Figur 74: D360 - Geiranger

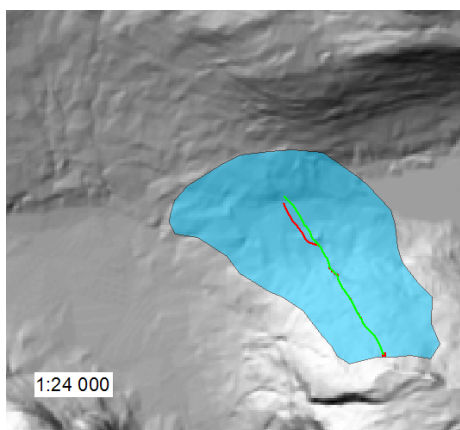
Figur 73, Figur 74 og Figur 75 viser sammenligning med et digitalisert skred. Her følger D360 skredet nøye, mens D3 har både en sikksakkete bane, og hvis startpunktet er identisk med startpunktet til skredet, avviker banen fullstendig.

Det er mulig å sette en vinkelbegrensning på D360 også, og med større nøyaktighet, siden D360 kan variere i mellom 0° og 360° . Det kan være synes intuitivt, ut ifra argumentet om skredets moment, og sette vinkelavviket lite. Dette kan i noen tilfeller gi tett sammenfallende resultater med de digitaliserte skredlinjene. Men i de fleste tilfeller viser en lav vinkel, som 15° seg å være for lite til å romme den dreiningen skredet har. Dette kan resultere i svært store avvik. Når algoritmen ikke greier å følge riktig bane, forsterkes feilen. Mer testing er nødvendig for å finne det ideale avviket for algoritmen, men forløpige tester indikerer at 45° , det samme avviket som for D3, gir gode resultater. Men muligens gir større tillat avvik, like trivielle forskjeller, som mellom D3 og D8.

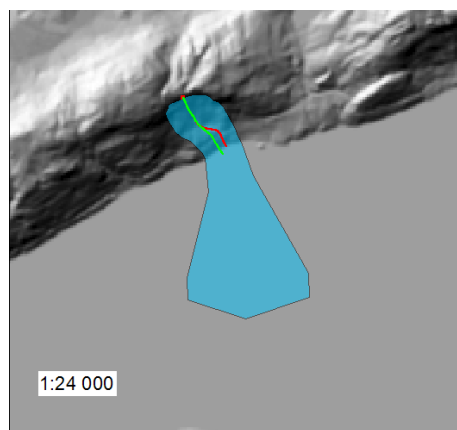


Figur 75: D3 (gul), D30 (rød) – Geiranger.

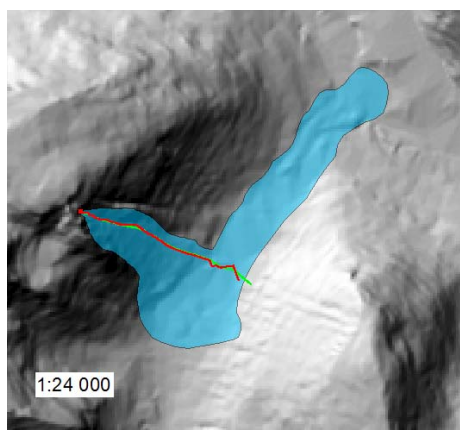
I sammenligningene med de skredene som ble digitalisert som polygoner, viser D360 og D3 ofte relativt likelydende resultater. To tilfeller er verdt å se nærmere på. Det ene er Skjærsura, hvor D360 opplever det samme avviket som D3 mot venstre (Figur 79).



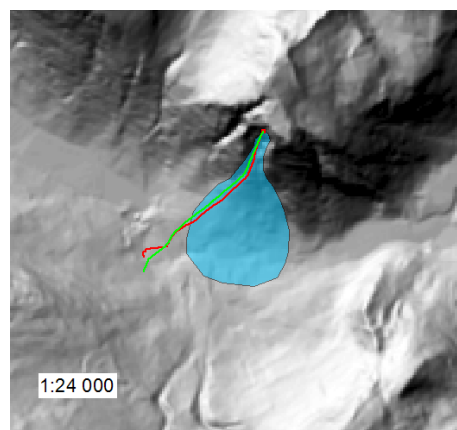
Figur 76: Byrkjelostøylen (D360)



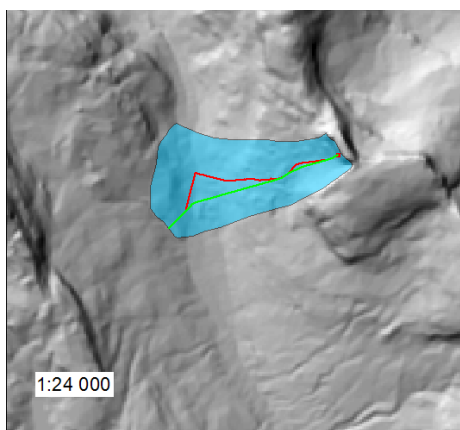
Figur 77: Tjelle (D360)



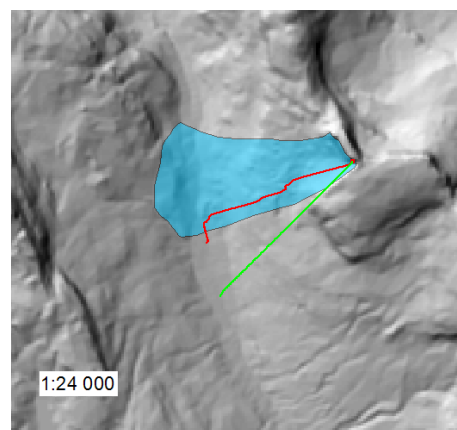
Figur 78: Kjusakdalen (D360)



Figur 79: Skjærsura (D360)



Figur 80: Gumpedalen A (D360)

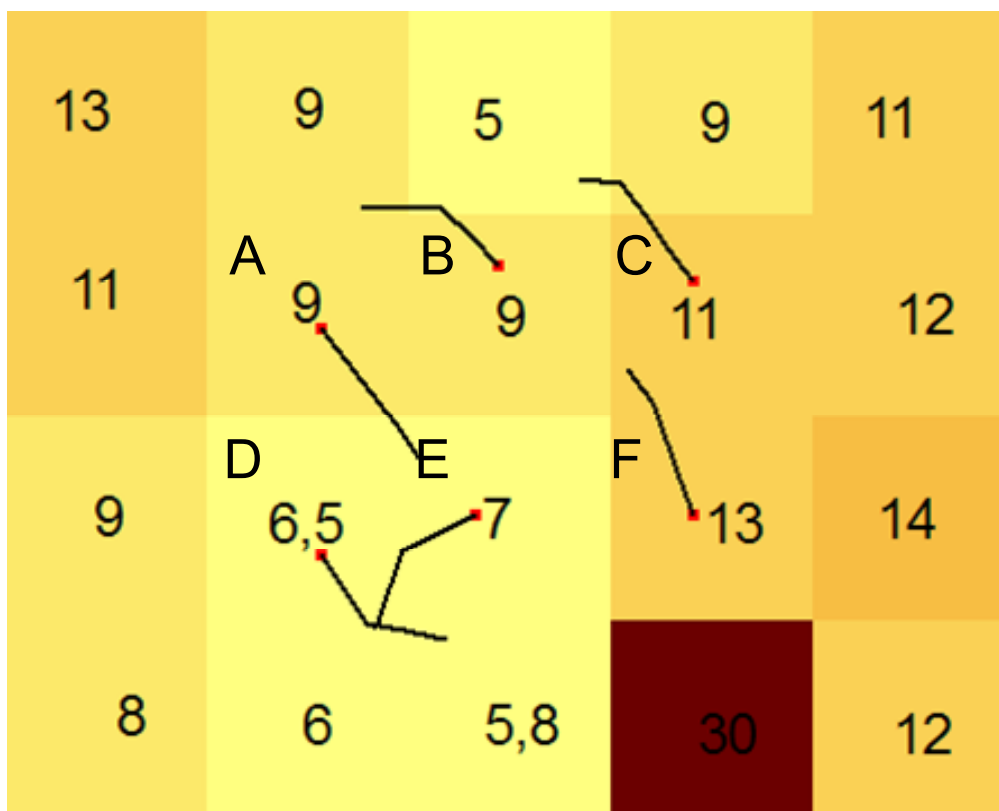


Figur 81: Gumpedalen B (D360)

Dette er ikke uventet når man retningen til D360 avgjøres av aspektet, og aspektet heller i retning sørvest (Figur 43). Men avviket fortsetter denne gangen i en mer realistisk retning. Det kan også se ut som om 45° avvik (rød strek) gir en urealistisk svingende effekt på vifta.

Det andre tilfellet er Gumpedalen. Ved valg av samme utgangspunkt som D3, avviker banen dramatisk, spesielt ved kun 15° tillat avvik (grønn) (Figur 80). Men flyttes startpunktet noe, gir det mer akseptable resultater (Figur 81).

Virkelige terrengmodeller gir altså ofte sammenfallende resultat, og dette styrker hypotese 3, da det ser ut til at D360 ikke gir analysene noe bedre resultat enn det som gis med D3. Noen dramatiske avvik finnes, og det er i D360s disfavør. Tillater man kun et avvik på 15° for D360 får man en jevnere bane enn ved 45°, men dette går på bekostning av at feil blir forsterket fordi algoritmen aldri greier å rette seg opp igjen, ved å svinge tilbake. Dette styrker hypotese 3 ytterligere, fordi valg av tillatt avvik er en kompliserende faktor.



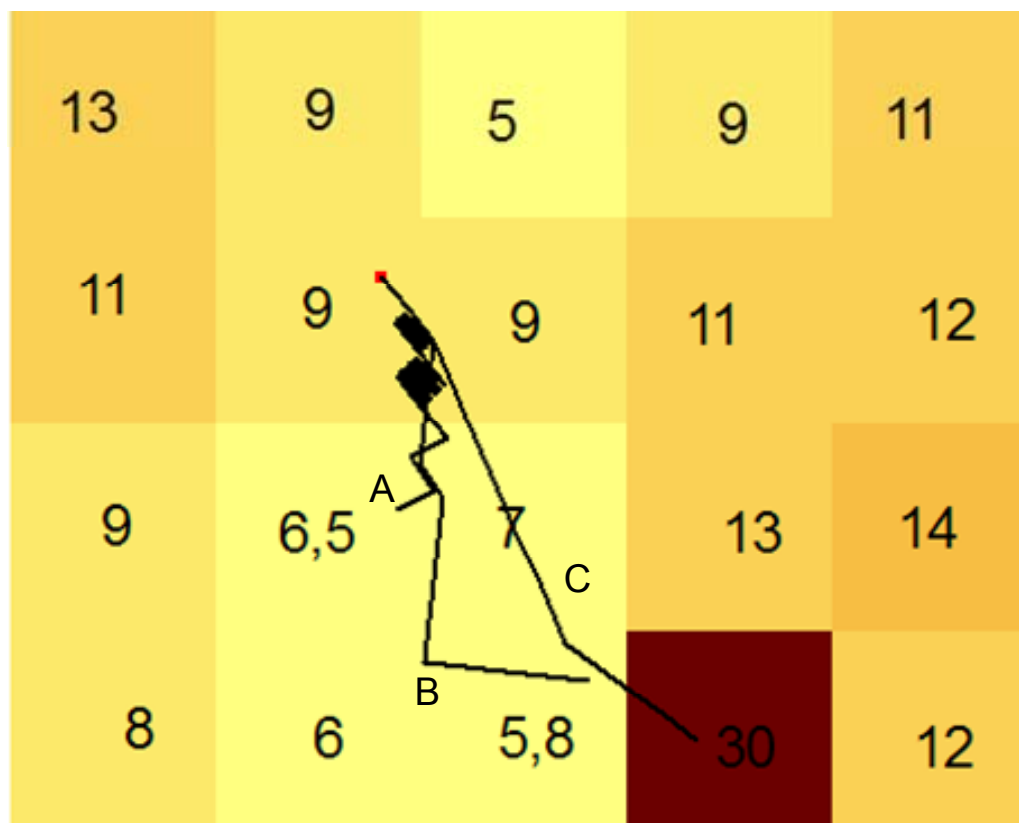
Figur 82: Sal-formet terrengmodell

Tarboton sammenligner sin egne algoritme Dinf, med 2D-Lea sin på en kunstig terrengmodell som er formet som en sal, dvs. at man kommer til et punkt hvor man har en helning i motsatte retninger (Tarboton 1997). Tarboton viser hvordan Leas algoritme, som er basert på å tilpasse et plan til hver celle, og dermed får en flytretning som minner om aspektet, får problemer med mange artefakter i en terrengmodell. Samme problemer hefter en annen

plantilpassningsalgoritme, DEMON (Costa-Cabral og Burges 1994). Figur 82 viser oppførselen til D360-algoritmen på en kunstig terrengmodell, identisk med den som er brukt av Tarboton.

I alle rutene bortsett fra E, har D360 sammen retning som Lea. Forskjellen skyldes måten flytretningen finnes på. I Leas algoritme skjer dette ved å tilpasse et plan til de fire hjørnepunktene til cellen. Disse igjen finnes ved å ta gjennomsnittet av høydene til de fire tilstøtende cellene. Flytretningen blir som en ball som ruller over dette planet, og denne retningen kan sies å være en måte å beregne aspektet. Aspektet er definert som angivelsen av retningen på den største helningen ut ifra et punkt, men alle måter å finne denne retningen på blir en tilnærming til virkeligheten. Måten å beregne aspektet på i D360 tar høyde for fire omkringliggende celler, mens Leas metode tar høyde for alle naboene.

I Leas tilfelle får man en loop mellom celle A og B fordi kanten mellom disse representerer et brudd på kontinuiteten i overflaten. Dette skyldes at de planene Lea tilpasser cellene blir tilnærminger, og ikke noen kontinuerlig flate. D360 har ikke de samme problemene



Figur 83: Forskjellig tillat avvik

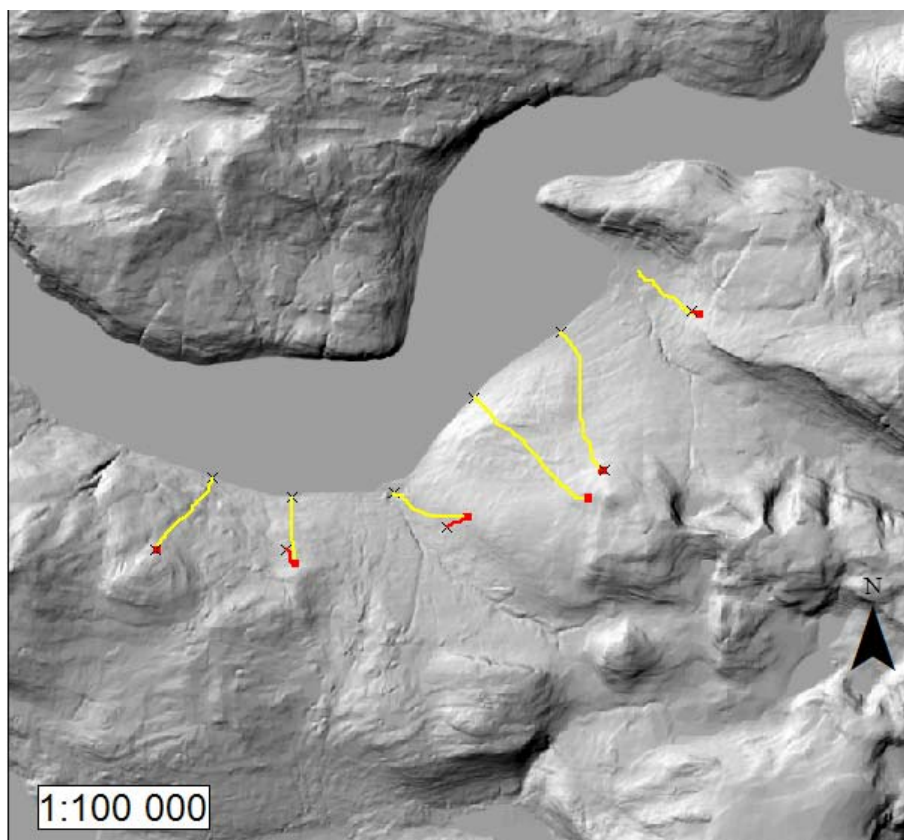
Alle forlater utgangspunktet i gal retning, men heller ikke videre blir resultatet riktig. I Figur 83 viser hvordan forskjellige vinkelbegrensninger av D360 gir forskjellige feil. Bane A når først inn i nabocellen, men nabocellen peker, som det ble vist i forrige figur (celle B), tilbake på utgangspunktet. Dermed oppstår en loop, hvor algoritmen krysser fram og tilbake mellom de to cellene. Denne banen har ingen vinkelbegrensning, og kan derfor snu 180°, og gå tilbake. Bane B og C, har derimot vinkelbegrensning, 45° og 15° respektivt. De har ikke mulighet å snu tilbake i den retningen de ”vil”. B nærmer seg denne retningen mest. Begge banene går mot cellen med høyde 30, pga ”edge contamination”, dvs. at kantcellene ikke gir riktig simulering.

Resultatene som er vist her og hos Tarboton, styrker hypotese tre, fordi de viser hvilke problemer som hefter mer avanserte algoritmer, inkludert D360.

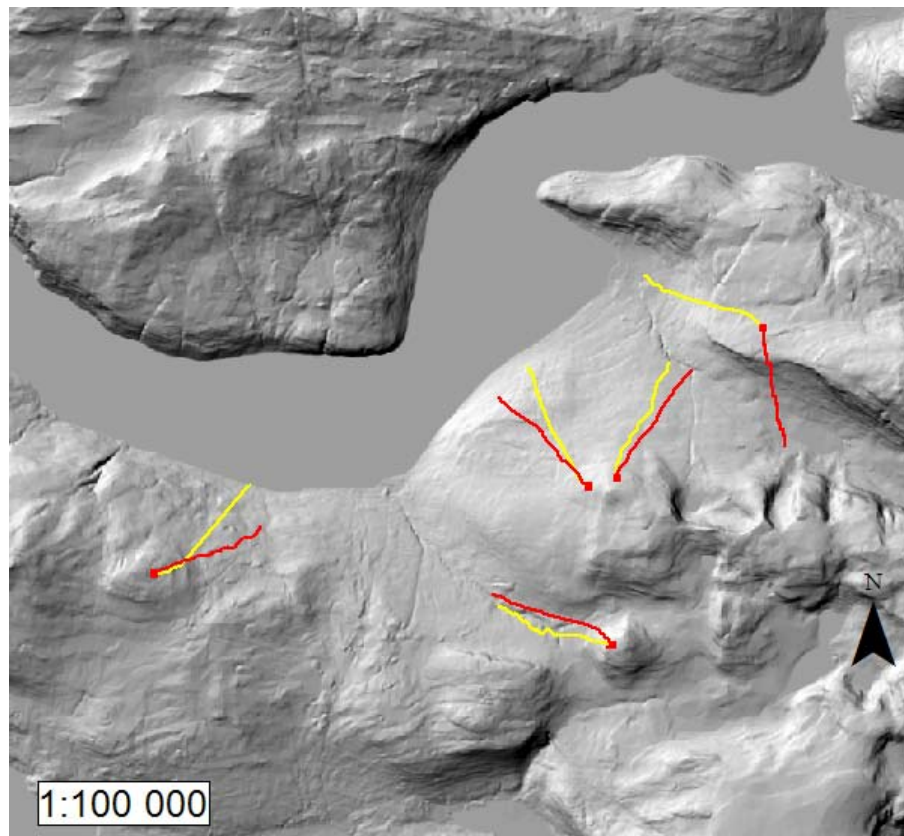
4.3.2 Sammenligning på en ASTER-basert terrengmodell

D3 og D360 ble også sammenlignet på en terrengmodell basert på ASTER-satellittbilder. Denne terrengmodellen viser en del av Innvikfjorden vest for Stryn. I Figur 84 er D3 kjørt henholdsvis på SKT25 og ASTER. D3 er ikke i stand til å gi brukbare resultater for ASTER-modellen, til tross for at en fyll-algoritme har blitt kjørt på denne. Som nevnt tidligere, blir alle pits fylt opp til samme høyde som laveste nabocelle. Det kan tenkes at resultatet hadde blitt mye bedre om man hadde tillatt algoritmen å bevege flyten til naboceller med samme høyde uansett i hvilken retning disse ligger.

Figur 85 viser tilsvarende sammenligning for D360-algoritmen. I dette tilfellet får man et bedre resultat, da denne lar seg i mindre grad påvirke av pits i landskapet. Dette viser en lang rekke avvikende resultater, samt et som har en hovedtrend mer enn 90° i forhold til retningen på SKT25. Selv om forskjellen i resultatene nok skyldes forskjeller i terrengmodellen, og i fire av fem tilfeller ikke er dramatiske, kan man lett få svært avvikende resultater. Denne tendensen til at en ASTER-terrengmodell gir avvikende resultater er også dokumentert i Sveits (Huggel et al. 2003b).



Figur 84: Innvikfjorden (D3). Gul (SKT), rød (ASTER).



Figur 85: Innvikfjorden (D30). Gul (SKT), rød (ASTER).

Disse resultatene representerer i en viss grad en svekkelse av hypotese tre. Det oppnåes ingen fornuftige resultater med D3, mens D360 kan gi gode resultater, selv om man kan få alvorlige feilskjær. Man kan dog innvende at siden terrengmodellen ikke er glattet, gir ikke dette den fulle oversikten. Resultatene her tyder dessuten på at D360 er mer velegnet til isolerte områder hvor man ikke har annet en satellittbilder å basere seg på (Kääb et al. 2003).

Man kan spekulere i egnetheten til de andre algoritmene som er presentert i teoridelen og som ikke er testet her. Mangelen på robusthet som D360 viste i simuleringene på "salen", gjelder også for DEMON og 2D-Lea (Tarboton 1997). Det er blitt vist at informasjon om skredbaner kan gå tapt hvis man ikke undersøker ties. Dette er en indikator på at MF-algoritmer som FD8, kan brukes. Men disse viser seg å gi for mye lateral spredning (Huggel et al. 2003b). Derfor er algoritmer som gir begrenset spredning å foretrekke. Slik sett er Dinf og D3R aktuelle for skredsimulering. Mange studier finner heftelser ved Rho8 (Costa-Cabral og Burges 1994, Tarboton 1997). Men den har ingen spredningsmodellering, gir ikke-reproduserbare resultater, og er muligens lite velegnet på virkelige terrengmodeller (Wilson et al. 2007). ut ifra dette sees hypotese tre styrket, men det kan ikke utelukkes helt at man kan profitere på mer avanserte algoritmer.

4.4 Hva er en modell?

I forbindelse med simulering av skred, brukes begrep som modell og algoritme. Begge deler er en abstraksjon av virkeligheten, som brukes for å nå et mål, det å si noe meningsfylt om en potensiell skredbane. Det er altså modellens evne til forutsigelse som man er interessert i. En simulert skredbane er som en hypotese som kan testes. Hvis hypotesen blir bekreftet, styrker det modellen, og i motsatt fall, hvis hypotesen avkreftes, svekker det modellen. Dette er i tråd med den hypotetisk deduktiv metode, som ligger til grunn for all vitenskap. Hvis modellens evne til forutsigelse minker, for eksempel ved at den stiller store krav til korrekte parametere og datakvalitet, reduseres bruksverdien. Det er derfor en omtrentlig modell som er anvendbare i alle sammenhenger, er å foretrekke fremfor en modell som er anvendbar kun i spesifikke tilfeller eller på spesifikke steder.

Man kan trekke et skille mellom fysiske og empiriske modeller (Dorren 2003). Fysiske er bygd opp av grunnleggende forståelse av fysiske lover, mens empiriske er basert på observerte resultater som er systematisert eller med andre ord, statistiske lovmessigheter. En

Topografisk modell faller mellom disse. Den suppleres riktignok med fordel fra enten fysiske eller empiriske modeller, for eksempel i forbindelse med avbruddskriterier for skred (Dorren 2003). Men den består ofte av en indre modell eller algoritme i programmet som produserer et resultat. Det er resultatet som avgjør modellens egnethet, ikke den indre algoritmen. Denne kan i teorien være tilfeldig designet. Det er egnetheten som er viktig. MSF-modellen er en kombinasjon av en "pathdistance" funksjon og en vekting som tillater et avvik på 45° fra bratteste skråning. Dette er begrunnet med at modellen skal simulere debris flow, som har en tendens til lateral spredning. Forklaringen er intuitiv, men ikke fysisk begrunnet (Huggel et al. 2003b). Debris flow som sådan er ikke et fenomen som beskrives ved at de beveger seg med høy sannsynlighet mot bratteste skråning, og med lineært minkende sannsynlighet vekk fra denne, og under ingen omstendigheter mer enn 45° til siden. Men denne algoritmen gir et resultat som stemmer bra overens med virkelige skred (Huggel et al. 2003b), og er derfor anvendbar.

I regional sammenheng er det mange tilnærminger man kan tillate seg å gjøre, som man ikke kan i lokal sammenheng. Kravet til presisjon er mindre, fordi man opererer på en større skala. Tilgjengeligheten av parametere er mindre, og kan ikke være avhengige av lokale forhold. Topografiske informasjon er derimot mer og mer universalt tilgjengelig. Kravet til enkel implementasjon er der, i forhold til tidsbruk. Selv om tilgjengelig datakraft øker generelt, øker også omfanget av data. Dermed må man fortsatt stille krav om enkel prosessering i regionale modeller. I regional kontekst er det både tillat og nødvendig å generalisere.

I en lokal kontekst må man ta hensyn til den underliggende fysikken. Modeller som baserer seg på forenklinger, blir rett og slett for upresise, som for eksempel Cone-modellen (Jaboyedoff og Labiouse 2003). Men i regional kontekst er tilnærminger og generaliseringer nødvendig og akseptabelt. Sammenligningen mellom D3, D8 og D360 viser dette. D3 har en begrensning på tillat avvik fra bratteste retning. Dette er ikke fysisk, men intuitivt begrunnet i skredets moment. Men denne begrensningen ser ut til å tilføre lite i forhold til D8, resultatmessig. På samme måte med D360. Det kunne være intuitivt å sette en begrensning på tillat avvik på 15°. Men dette førte til uventede og uønskede resultater. 45° ble valgt, fordi det så ut til å passe bedre.

I alle disse sammenhengene er man avhengig av data for ekte skred til evaluering. Dette er under forutsetningen at også skred som ikke har inntruffet er fenomener som deler egenskaper

med de vi har kartlagt. Flere skred er kartlagt som har inntruffet i førhistorisk tid. Men også disse kan brukes til evaluering, siden man regner med at forholdene ikke har forandret seg nevneverdig.

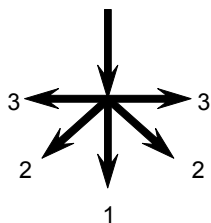
Fremgang på dette feltet er avhengig av at så mye som mulig og så nøyaktig som mulig kartlegging av skred. Disse brukes så til å kalibrere modellene. Så kan man tenke seg en modell som automatiserer variasjon av algoritmene og finner de best egnede. Den algoritmen som så er resultatet har kanskje ingenting med den fysiske virkeligheten å gjøre, men er likevel velegnet fordi den er treffsikker.

Det bør også nevnes at terrengmodellene også er tilnærminger av virkeligheten. Måten informasjonen i terrengmodellen brukes på varierer jo fra D8, hvor høyden er antatt sentrert i cellen, til plantilpasningsalgoritmene DEMON og 2D-Lea, til Dinf, som med sine fasetter representerer en tilnærming til en TIN-modell. Disse kan også være mer eller mindre velegnet. En flytalgoritme som kjøres på et grid er i så måte en modell som utføres på en annen modell. Det er påpekt at inkonsistente resultater kan oppstå som følge av at disse to modellene generaliserer virkeligheten på forskjellig måte (Schneider 2001). Mer innsikt i hvordan man kan få disse to generaliseringene av virkeligheten mer samstemte, er påkrevd.

5 Konklusjon

En flytalgoritme kalt D3 ble implementert og resultatene av kjøringer sammenlignet med kjente skred, registrert både som linjer og polygoner. Disse sammenligningene viste noen avvik, men generelt en god match. Begrensninger som ble påvist er manglende mulighet til å simulere skred som passerer terreng som skråner oppover. Det ble også påvist at begrensningen til åtte hovedretninger påvirker resultatet negativt, samt utilstrekkelig oppførsel på konvekse flater (Skjærsura). Det ble påvist at noen skred lot seg utilstrekkelige simulere (Kjusakdalen). Et avbruddskriterium basert på forholdet h/l var implementert, og det ble funnet at $h/l=0,02$ var et altfor romslig kriterium. Høyde null fungerte som barriere i tillegg, for å hindre at simuleringen fortsatt ut i fjorden. Det ble påvist at motivasjonen for å begrense tillat avvik fra forrige retning, med begrunnelse i skredets moment, kun førte til trivielle forskjeller. Dette ble påvist ved å sammenligne en egenimplementert D8 med D3. Simuleringen, både for D3 og D8, resulterte ofte i pits, selv på terrengmodeller som var prosessert med en fyll-algoritme. I begge tilfeller tillates bare bevegelse rett forover ved lik høyde.

Sammenligningene mellom D3 og ekte skred viser at algoritmen kan gi brukbare resultater i en regional kontekst. Negative aspekter ved algoritmen reduserer kvaliteten på resultatet. Hypotese en sees således bekreftet, med det forbehold at det er lite å vinne på å bruke D3 framfor D8. Resultatene viser at algoritmen bør tillate flyt til alle naboceller med lik høyde, men at man kan ta hensyn til momentet ved skredet ved å innføre et hierarki på retningen som sjekkes. Først sjekkes de retningene som medfører minst avvik. Hvis og bare hvis ingen gyldig bane finnes, sjekkes så naboene som ligger 90° til siden (Figur 86). Dette bør så kjøres på en terrengmodell som har blitt prosessert med en fyll-algoritme.



Figur 86: Hierarki på retningene som sjekkes

Det ble så utført en analyse av flere terrengmodeller med en algoritme som undersøkte forekomsten av ties, dvs. flertydighet i valg av retning. Konseptet om terskelverdi for ties ble innført. Dette innebærer at man registrer ties selv om verdiene ikke er eksakt like, men innenfor et slingringsmonn satt av terskelverdien. Denne terskelverdien kan enten være absolutt eller relativ i forhold til helningsvinkelen. En analyse ble gjort på en terrengmodell fra Geirangerfjorden med terskelverdi på henholdsvis 0, 0,02 og 0,05. Et området fra Jostedalsbreen ble undersøkt både for en SKT25 og en ASTER-terrengmodell. Resultatene viste at det forekom et ikke ubetydelig antall ties selv ved terskel null, at dette antall økte med terskelverdiene og at de varierte, med noen unntak uniformt i terrenget. Første del av hypotese to er bekreftet. Det ble påvist mange flere ties for ASTER-terrengmodellen, og dette ble satt i sammenheng med at den ikke er glattet.

Det ble vist at konsekvensen av ties kan både være divergerende og konvergerende og at divergerende tilfeller kan gi betydelige forskjeller. En variant av D3, D3R ble implementert for å teste konsekvensene. Det viste seg at ved en terskelverdi på 0,02 forekom ties hyppig og at de gav et mer utfyllende bilde av mange skred. Det ble også observert at hvis tie-verdien ble satt tilstrekkelig høyt, kunne D3R fungere som en multiple flyt-algoritme og gi et inntrykk av skredets mulig utbredelse. Andre del av andre hypotese sees ikke bekreftet, siden det forekommer divergerende baner. Det bemerkes at hvis man tillater en viss terskelverdi og simulerer banene fra alle tie-punktene, får man potensielt mer informasjon om skredets mulige baner og dets utbredelse.

D3 ble sammenlignet med en algoritme som baserer seg på aspektet for å beregne flytretning, kalt D360. Algoritmene ble sammenlignet på plane flater, konvekse og konkave flater i tillegg til terrengmodeller med forskjellige opphav og med virkelige skred. D3 oppviste mange kjente begrensning på de konstruerte flatene, mens D360 gav akseptable resultater. Det ble påvist at D360 kunne gi gode resultater på en terrengmodell med dårlig kvalitet (ASTER), men det bemerkes at sammenligningen ikke er fullstendig før D3 er implementert på en måte som er skissert ovenfor. Så langt er hypotese tre svekket. D360 viste akseptable resultater sammenlignet med virkelige skred. Men det viste seg at den kan gi mer markante feil enn D3. Feilene ble i tillegg forsterket hvis det ble satt en begrensning på avvik fra forrige retning. Selv om en slik begrensning glattet resultatet og gav en mer realistisk skredbane, førte det også til at begynnelsen på en feil bane aldri ble rettet opp. Forsøk med en terrengmodell formet som en sal viste dessuten at D360 har flere problemer, spesielt hvis tillat avvik settes

høyt. Den blir for påvirket av høye naboer. Hypotese tre er derfor styrket. I tillegg bekreftes den ved eksempler fra andre studier, og ved det var lite å hente på å legge en kompliserende faktor til D8 for å få D3. Begrensningene ved Skjærsura og det at ingen av algoritmene tillater skred å passere stigende terreng, bidrar til å svekke hypotese tre. Totalt sett sees hypotese tre bekreftet.

Skredmodellering basert på flytalgoritmer gir størst nytte på regionalt nivå. I lokal sammenheng er de i beste fall et supplement. Et akseptabelt resultat oppnåes ved å kombinere en GIS-modell med en empirisk modell for skredlengden. Det trenger ikke være fysiske lovmessigheter som ligger til grunn for algoritmen, så lenge resultatene er verifiserbare i forhold til virkelige skred. En simulering langs en linje gir mindre informasjon om utbredelsen av skredet, men åpner for mange muligheter for mer avanserte beregninger. Utbredelsen av skredet kan simuleres ved bruk av D3R.

Sammenligningene i oppgaven mellom ekte skred og simulerte kunne være gjort mer presist ved at startpunktet ikke ble valgt manuelt. For skredene som er digitalisert som polygoner, kan man stille spørsmål ved valg av utgangspunkt. Videre er det en svakhet at ties ble beregnet i to forskjellige programmeringsmiljøer.

5.1 Videre arbeid

Algoritmen kan utvides slik det skisseres ovenfor, med hierarkisk sjekking av retningene. Kanskje også andre vekteregimer er mulige. Dette vil en bedre mulighet til å sammenligne både med virkelige skred, terrengmodeller med dårlig kvalitet og med andre algoritmer. Det kan også implementeres en automatisk sjekk av alle punkter i et løseområde. Videre kan andre mekanismer for avbrudd, for eksempel basert på kinetisk energi benyttes (Formel 6). Det kan også undersøkes nøyere hvilke terrengtyper som frembringer mest ties. D3R-algoritmen kan forbedres ved at også avbruddsmekanismer håndteres og at ikke overlappende stier undersøkes. D3Rs muligheter til å simulere multiple flyt kan undersøkes videre. Potensialet til Dinf bør undersøkes.

6 Kilder

- Bakkehøi, B., Harbitz, C.B. og Kveldsvik, V. 2004. *Hotell Og Glassbjelke Ved Zakariasdemningen*. Norges Geotekniske Institutt (NGI).
- Berry, J.K. *Manual of Geographic Information Systems*. University of Denver 2008 [nedlastet 01.03.2008. Tilgjengelig fra <http://www.innovativegis.com/basis/Papers/Other/ASPRChapter/>.
- Blikra, L.H. 2003. Tsunamis Generated by Rockslides in Geiranger- and Tafjorden - Scenarios and Model Comparisons. *Norges Geotekniske Institutt (NGI)*.
- Braathen, A., Blikra, L.H., Berg, S.S. og Karlsen, F. 2004. Rock-Slope Failures in Norway; Type, Geometry, Deformation Mechanisms and Stability. *Nowegian Journal of Geology* 84, 67-88.
- Burrough, P.A. og McDonnell, R.A. 1998. *Principles of Geographical Information Systems*, Oxford: Oxford University Press.
- Costa-Cabral, M. og Burges, S.J. 1994. Digital Elevation Model Networks (Demon): A Model of Flow over Hillslopes for Computation of Contributing and Dispersal Areas. *Water Resources Research* 30, 1681-1692.
- Dahl, O.J. og Nygaard, K. 1966. Simula: An Algol-Based Simulation Language. *Communications of the ACM archive* 9 (9), 671-678.
- Dorren, L.K.A. 2003. A Review of Rockfall Mechanics and Modelling Approaches. *Progress in Physical Geography* 27 (1), 69-87.
- Arcinfo 9.2. Environmental Systems Research Institute Inc. (ESRI), Redlands, CA., USA.
- Etzelmüller, B. og Sulebak, J.R. 2000. Developments in the Use of Digital Elevation Models in Periglacial Geomorphology and Glaciology.
- Fairfield, J. og Leymarie, P. 1991. Drainage Networks from Grid Digital Elevation Models. *Water Resources Research* 27 (5), 709-717.
- Goodchild, M.F. *Lecture 13: Analysis 3*. University of California 2008 [nedlastet 01.03.2008. Tilgjengelig fra <http://www.geog.ucsb.edu/~good/176b/m13.html>.
- Guzzettia, F., Crostab, G., Dettic, R. og Agliardid, F. 2002. Stone: A Computer Program for the Three-Dimensional Simulation of Rock-Falls. *Computers & Geosciences* 28, 1079-1093.
- Harbitz, C.B. 2002. *Ormen Lange – Ilandføringsalternativer: Rock Slide Generated Tsunamis - Run-Up Heights at Baraldsnes and Nyhamn*. 16 s.
- Harbitz, C.B., Løvholt, F. og Blikra, L.H. 2003. *Tsunamis Generated by Rockslides in Geiranger- and Tafjorden - Scenarios and Model Comparisons*. Norges Geotekniske Institutt (NGI). 14 s.

- Hausberg, J.E. 2008. *Upublisert*, Institutt for Geofag, Universitetet i Oslo, Oslo.
- Hestnes, E. 1979. *Kartlegging Av Skredfarlige Områder I Norge*. Norges Geotekniske Institutt (NGI). 26 s.
- Hestnes, E. 1985a. *Hvilke Skredtyper Må Planleggerne Ta Hensyn Til?* Norges Geotekniske Institutt (NGI). 31 s.
- Hestnes, E. 1985b. *Skredfarekart - Et Hjelpemiddel Ved Arealplanlegging*. Norges Geotekniske Institutt (NGI). 13 s.
- Huggel, C., Haeberli, W., Kääb, A., Hoelzle, M., Ayros, E. og Portocarrero, C. 2003a. Assessment of Glacier Hazards and Glacier Runoff for Different Climate Scenarios Based on Remote Sensing Data: A Case Study for a Hydropower Plant in the Peruvian Andes. Paper read at Proceedings of EARSeL-LISSIG-Workshop; Observing our Cryosphere from Space, 11.03.2002, at Bern, Switzerland.
- Huggel, C., Kääb, A., Haeberli, W. og Krummenacher, B. 2003b. Regional-Scale Gis-Models for Assessment of Hazards from Glacier Lake Outbursts: Evaluation and Application in the Swiss Alps. *Natural Hazards and earth System Sciences* 3, 647-662.
- Huggel, C., Kääb, A. og Salzman, N. 2004. Gis-Based Modeling of Glacial Hazards and Their Interactions Using Landsat-Tm and Ikonos Imagery. *Norwegian Journal of Geography* 58, 61-73.
- Hugget, R. og Cheesman, J. 2002. *Topography and the Enviroment*, London: Longman Group United Kingdom.
- Iverson, R.M. og Schilling, S.P. 1998. Objective Delineation of Lahar-Inundation Hazard Zones. *GSA Bulletin* 8, 972-984.
- Jaboyedoff, M. 2003. *Conefall 1.0: User's Guide*. Quanterra - International Independent Center of Climate Change. 15 s.
- Jaboyedoff, M. og Labiouse, V. 2003. Preliminary Assessment of Rockfall Hazard Based on Gis Data. Paper read at ISRM 2003–Technology roadmap for rock mechanics, 08.09.2003, at Sandton Convention Centre, Gauteng, South Africa.
- Kääb, A., Huggel, C., Paul, F., Wessels, R., Raup, B., Keiffer, H. og Kargel, J. 2003. Glacier Monitoring from Aster Imagery: Accuracy & Applications. Paper read at Proceedings of E ARSeL-LISSIG-Workshop Observing our C ryosphere from Space, at Bern.
- Matlab R2007b*, Natick, Boston, USA.
- Meissl, G. 1998. *Modellierung Der Reichweite Von Felsstürzen – Fallbeispiele Zur Gis-Gestützten Gefahrenbeurteilung Aus Dem Bayrischen Und Tiroler Alpenraum*, Innsbrucker Geographische Studien, Innsbruck.

- Metternicht, G., Hurni, L. og Gogu, R. 2005. Remote Sensing of Landslides: An Analysis of the Potential Contribution to Geo-Spatial Systems for Hazard Assessment in Mountainous Environments. *Remote Sensing of Enviroment* 98, 284-303.
- Microsoft. Vba. Microsoft 2008 [nedlastet 01.03.2008. Tilgjengelig fra <http://msdn2.microsoft.com/en-us/isv/bb190538.aspx>.
- O'Callaghan, J.F. og Mark, D.M. 1984. The Extraction of Drainage Networks from Digital Elevation Data. *Computer Vision Graphics and Image Processing* 28, 323-344.
- Quinn, P.F., Beven, K.J., Chevallier, P. og Planchon, O. 1991. The Prediction of Hillslope Paths for Distributed Hydrological Modelling Using Digital Terrain Models. *Hydrological Processes* 5, 59-79.
- Ramberg, M. 2007. *Interferometrisk Nedsynkningsanalyse Og Superoppløsning Av Sar Satellittbilder*, Institutt for Geofag, Universitetet i Oslo, Oslo.
- Romstad, B. 2001. *Automatisk Landformkartlegging Med En Kontekstuell Relieffklassifisering*, Institutt for Geofag, Universitetet i Oslo, Oslo.
- Romstad, B., Blikra, L.H. og Etzelmüller, B. 2006. Norwegian Lakes Exposed to Tsunami Generating Rock Slides. *Upublisert*.
- Salzmann, N., Kääh, A., Huggel, C., Allgöwer, B. og Haeberli, W. 2004. Assessment of the Hazard Potential of Ice Avalanches Using Remote Sensing and Gis-Modeling. *Norwegian Journal of Geography* 58 (74-84).
- Scheidegger, A.E. 1973. On the Prediction of the Reach and Velocity of Catastrophic Landslides. *Rock Mechanics* 5, 231-236.
- Schneider, B. 2001. Phenomenon-Based Specification of the Digital Representation of Terrain Surfaces. *Transactions in GIS* 5 (1), 39-52.
- Tarboton, D.G. 1997. A New Method for the Determination of Flow Directions and Upslope Areas in Grid Digital Elevation Models. *Water Resources Research* 33 (2), 309-319.
- Tarboton, D.G. 2003. Terrain Analysis Using Digital Elevation Models in Hydrology. *23rd ESRI International Users Conference*, 07.11.2003, San Diego, California, USA.
- Terrain Analysis Using Digital Elevation Models*. Utah State University
- Turner, K. og Schuster, R.-. 1996. *Landslides - Investigation and Mitigation*, U.S.: Transportation Research Board.
- Wilson, J.P. og Gallant, J.C. 2000. *Terrain Analysis: Principles and Applications*, New Jersey: Wiley.
- Wilson, J.P., Lam, C.S. og Deng, Y. 2007. Comparison of the Performance of Flow-Routing Algorithms Used in Gis-Based Hydrologic Analysis. *Hydrological Processes* 21, 1026-1044.

Appendiks

D3R - Slidetree

```
Public Function slidetree(ByRef pRasterBand As IRasterBand, _
    ByRef pPoint As IPoint, _
    Optional ByVal MaxAngle As Integer = 1, _
    Optional ByVal Recursive = False, _
    Optional ByVal trueD8 = False)

    Dim pCheckpoints As IPointCollection
    Dim pFirstPath As IPointCollection
    Dim pStartPoint As IPoint
    Dim pTempPath As IPointCollection
    Dim intTieCount As Integer
    Dim blFirst As Boolean
    Dim pVisited As IPointCollection
    Dim pFoo As IPoint
    Dim blVisited As Boolean

    ' error checking
    If Not TypeOf pPoint Is IPoint Then
        MsgBox "Error: Illegal point", vbCritical
    End If
    ' 0 = 0, 1 = 45, 2 = 90, 3 = 135 degrees
    If MaxAngle < 0 Or MaxAngle > 3 Then
        MsgBox "Error: Maxdegree value" & MaxAngle, vbCritical
    End If
    If Not TypeOf pRasterBand Is IRasterBand Then
        MsgBox "Error: Illegal rasterband. Context Slide", vbCritical
    End If

    ' initialize all variables
    Set pMxDoc = ThisDocument
    Set pActiveView = pMxDoc.FocusMap
    Set pScreenDisplay = pActiveView.ScreenDisplay
    Set pStartPoint = pPoint
    Set pCheckpoints = New path
    Set pVisited = New path
    SetRasterBand pRasterBand
    intMaxSteps = frmOptions.txtMaxStepsD8
    row = 0
    col = 0
    intMaxAngle = MaxAngle
    blRecursive = Recursive
    blTrueD8 = trueD8
    blVisited = False
    ' Set up the 3x3 window with direction objects
    Dim intI, intJ As Integer
    For intI = 0 To 2
        For intJ = 0 To 2
            Set pCell(intI, intJ) = New Direction
        Next
    Next
    'sett opp neste og forrige pekere
    setMapDirections
    'SetRC kalles her i tillegg til i slidemetoden, delvis som en errorsjekk på
    'startpunktet, og delvis fordi SetXY kalles etterpå og bruker row/col
    If Not SetRC(pStartPoint) Then
        'If the point is outside grid bounds exit sub
        MsgBox "Error: Point is outside grid boundaries", vbCritical
    End
    End If
    ' SetXY basically sets x,y coordinates, but also centers them in the cell for
drawing
    SetXY pStartPoint
    DrawPoint pStartPoint, 255, 0, 0, 1, 3
```

```

pStartPoint.z = 0
If Not Recursive Then
' Hvis ikke rekursivitet ønskes, kjøres en gang før exit
Set pFirstPath = Slide(pStartPoint, True)
DrawPolyLine pFirstPath, frmOptions.txtR, frmOptions.txtG, _
frmOptions.txtB, frmOptions.txtLineWidth
slidetree = pFirstPath
Exit Function
End If
' Add the startpoint to checkdirection even though it might not be a tie, to
start mainloop
pCheckpoints.addPoint pStartPoint

blFirst = True
Dim i, j As Integer
' Main loop for finding all ties
Do While (Not pCheckpoints.PointCount = 0)
'tar hele tiden fra første
Set pTempPath = Slide(pCheckpoints.Point(0), blFirst)
If blFirst Then
Set pFirstPath = pTempPath
End If
' loop through path for ties (z value not zero)
For i = 0 To pTempPath.PointCount - 1
If pTempPath.Point(i).z <> 0 Then
'sjekker for duplikater
For j = 0 To pVisited.PointCount - 1
If pVisited.Point(j).Compare(pTempPath.Point(i)) Then
blVisited = True
Exit For
End If
Next j
If Not blVisited Then
' nye punkter legges på til slutt
pCheckpoints.addPoint pTempPath.Point(i)
End If
blVisited = False
End If
Next i
blFirst = False
' Metoden med å fjerne duplikater fungerer ikke etter hensikten
'pVisited.AddPoint pCheckpoints.Point(0)
pCheckpoints.RemovePoints 0, 1
'MsgBox "hello"
If frmOptions.cbRnd Then
' random color
DrawPolyLine pTempPath, Int((256) * Rnd), Int((256) * Rnd), _
Int((256) * Rnd), frmOptions.txtLineWidth
Else
DrawPolyLine pTempPath, frmOptions.txtR, frmOptions.txtG, _
frmOptions.txtB, frmOptions.txtLineWidth
End If
intTieCount = intTieCount + 1
If intTieCount = 100 Then
Exit Do 'en bug forårsaker mange ties; avbruddsmekanisme
MsgBox "Ties: " & intTieCount
End If
Loop
'MsgBox "Ties: " & intTieCount
slidetree = pFirstPath

End Function

```

D3: Slide

```
Private Function Slide(ByRef pPoint As IPoint, blStart As Boolean)
```

```

Dim pP As IPoint
Dim pPathPoints As IPointCollection
Dim intDir As Integer
Dim intDirOld As Integer
Dim dblH As Double
Dim dblL As Double
Dim dblStartZ As Double

Set pPathPoints = New path
'get the input coordinates into local variable
Set pP = New Point
pP.PutCoords pPoint.x, pPoint.y
pP.z = pPoint.z
'Set slidegrid row and col to current point position
SetRC pP
' Reset. At this point we start a new path. For the sub-starting
' points of previous ties the getflowdir is skipped, so the old
' direction is not needed.
intDirOld = 0

' Main loop; i changed to start at 1.
Dim i As Integer
For i = 0 To intMaxSteps
    'read in z values into window (based on row/col)
    readWin
    ' hent starthøyde
    If blStart Then dblStartZ = pCell(1, 1).z
    ' Get the new direction
    If pP.z = 0 Then
        intDir = getflowdir(intDirOld, blStart, pP)
    ElseIf pP.z > 255 Or pP.z < 0 Then
        MsgBox "Illegal intDir: " & intDir, vbCritical
    Else
        ' In case of earlier tie, skip getflowdir
        intDir = pP.z
    End If
    ' Decide what to do with the direction output
    If intDir = 0 Then
        'pit
        'add the last point (the pit point)
        pPathPoints.addPoint pP
        ' mark pitpoint with cross
        DrawPoint pP, 0, 0, 0, 3, 5
        Slide = pPathPoints
        Exit Function
    ElseIf (intDir = 1 Or intDir = 2 Or intDir = 4 Or intDir = 8 Or intDir = 16
Or intDir = 32 Or intDir = 64 Or intDir = 128) Then
        ' Non-tie
        ' reset z
        pP.z = 0
    ElseIf intDir < 0 Or intDir > 255 Then
        ' Just in case
        MsgBox "Illegal intDir: " & intDir, vbCritical
    Else
        ' Tie
        If (Int(intDir / 128)) > 0 Then
            pP.z = (intDir - 128) 'remaining directions
            intDir = 128 'chosen direction
        ElseIf (Int(intDir / 64)) > 0 Then
            pP.z = (intDir - 64) 'remaining directions
            intDir = 64
        ElseIf (Int(intDir / 32)) > 0 Then
            pP.z = (intDir - 32) 'remaining directions
            intDir = 32
        ElseIf (Int(intDir / 16)) > 0 Then
            pP.z = (intDir - 16) 'remaining directions
            intDir = 16
        ElseIf (Int(intDir / 8)) > 0 Then

```

```

        pP.z = (intDir - 8) 'remaining directions
        intDir = 8
    ElseIf (Int(intDir / 4)) > 0 Then
        pP.z = (intDir - 4) 'remaining directions
        intDir = 4
    ElseIf (Int(intDir / 2)) > 0 Then
        pP.z = (intDir - 2) 'remaining directions
        intDir = 2
    ElseIf (Int(intDir / 1)) > 0 Then
        pP.z = (intDir - 1) 'remaining directions
        intDir = 1
    End If
    ' Draw the tiepoints
    If frmOptions.cbDisplayTies Then
        If frmOptions.cbRndTieBox Then
            DrawPoint pP, _
                Int((256) * Rnd), _
                Int((256) * Rnd), _
                Int((256) * Rnd), 4, 5
        Else
            DrawPoint pP, 0, 255, 0, 4, 5
        End If
    End If
    ' add point to path
    pPathPoints.addPoint pP
    ' reset tiepoint
    pP.z = 0
    'update row and col to steepest downslope cell (remember that row is
TopLeftCorner)
    Select Case intDir
        Case 1 ' NW
            row = row - 1
            col = col - 1
        Case 2 ' N
            row = row - 1
            col = col
        Case 4 ' NE
            row = row - 1
            col = col + 1
        Case 8 ' W
            row = row
            col = col - 1
        Case 16 ' E
            row = row
            col = col + 1
        Case 32 ' SW
            row = row + 1
            col = col - 1
        Case 64 ' S
            row = row + 1
            col = col
        Case 128 ' SE
            row = row + 1
            col = col + 1
        Case Else
            MsgBox "Error!", vbCritical
    End Select
    ' Avbruddshåndtering (forutsetter dx og dy er like store for
enkelhetsskyld)
    ' Ved første punkt skal ingen lengde telles
    ' NB skill mellom blStart og blD8
    If frmOptions.cbHLYesNo And Not blStart And Not blRecursive Then
        If getDirPointer(intDirOld).blDiagonal Then
            dblL = dblL + dx * Sqr(2)
        Else
            dblL = dblL + dx
        End If
    End If

```

```

        dblH = dblStartZ - pCell(1, 1).z
        If (dblH / dblL) < CDbl(frmOptions.txtHLRatio) Then
            MsgBox "Avbrudd etter: " & Round(dblL, 2) & " m." & vbNewLine _
                & "Høyde: " & dblH & vbNewLine & "H/L: " & dblH / dblL
            Exit For
        End If
    End If
    ' ved trued8 sjekkes alltid alle retninger, som ellers ved startpunkt
    If Not blTrueD8 Then blStart = False
    ' Ta vare på forrige retning
    intDirOld = intDir
    ' Update the coordinates in pP point based on new row/col values
    SetXY pP

Next
Slide = pPathPoints

End Function

```

D360: SlidePath360

```

Public Function SlidePath360(ByRef pPoint As IPoint, _
    Optional ByVal dblStepLength As Double = 5, _
    Optional ByVal dblMaxAngle As Double = 5, _
    Optional ByVal intSteps As Integer = 5) As IPointCollection
    'MaxAngle in degrees

    dblMaxAngle = dblMaxAngle * PI / 180

    'First find the row and col of point pP
    If Not SetRC(pPoint) Then
        'If the point is outside grid bounds exit sub
        MsgBox "Error: Point is outside grid boundaries"
        Exit Function
    End If

    Dim dblTopZ As Double
    Dim dblZ As Double
    Dim dblLen As Double
    Dim pP As IPoint
    Set pP = New Point

    pP.PutCoords pPoint.x, pPoint.y

    'Define a point collection
    Dim pPathPoints As IPointCollection
    Set pPathPoints = New path

    Dim varSurfaceArray As Variant
    'Get the local surface
    varSurfaceArray = GetLocalWindow()
    dblTopZ = varSurfaceArray(1, 1)
    dblZ = dblTopZ
    dblLen = 0

    Dim dblAspect As Double, dblOldAspect As Double
    Dim dblSlope As Double, dblOldSlope As Double
    Dim dblAspectDiff As Double
    Dim dblNewX As Double, dblNewY As Double
    Dim i As Integer
    dblOldAspect = getAspectnew(varSurfaceArray)
    dblOldSlope = getAspectnew(varSurfaceArray)
    i = 0
    Do While i < intSteps
        'Add point to line
        pPathPoints.addPoint pP
        varSurfaceArray = GetLocalWindow()
        'Get aspect and slope from local window

```

```

    dblAspect = getAspectnew(varSurfaceArray)
    dblSlope = getSlope(varSurfaceArray)

    If dblAspect = -2 Then
        MsgBox "Aspect calculation failed"
        Exit Do
    ElseIf dblAspect = -1 Then
        dblAspect = dblOldAspect
    End If
    'Make sure direction doesn't change more than specified
    dblAspectDiff = dblAspect - dblOldAspect

    If Abs(dblAspectDiff) > dblMaxAngle Then
        dblAspect = dblOldAspect + (dblMaxAngle * dblAspectDiff /
Abs(dblAspectDiff))
    End If

    ' changed cos and sin because aspect is angle from north
    dblNewY = Cos(dblAspect) * dblStepLength
    dblNewX = Sin(dblAspect) * dblStepLength

    'DrawPoint pP, 0, 0, 250, 0, 6

    'Move the point to this location
    pP.PutCoords pP.x + dblNewX, pP.y + dblNewY
    If Not SetRC(pP) Then
        'If the point is outside grid bounds exit sub
        MsgBox "Path moved out of bounds"
        Exit Do
    End If
    dblAspect = getAspectnew(varSurfaceArray)
    dblSlope = getAspectnew(varSurfaceArray)
    i = i + 1
Loop
'Return the point collection
Set SlidePath360 = pPathPoints
End Function

```

D360: getaspect

```

Public Function getAspect(ByRef myArray As Variant) As Double
    'Returns aspect (in radians) for the local window

    If Not IsArray(myArray) Then
        'No surface in input, return -2 (error)
        getAspect = -2
        Exit Function
    End If

    Dim i As Integer
    For i = 0 To 8
        If myArray(dj(i) + 1, di(i) + 1) = dblNodata Then
            'If any cell is nodata, set aspect to nodata
            getAspect = dblNodata
            Exit Function
        End If
    Next i

    Dim z1 As Double, z2 As Double, z3 As Double, z4 As Double, z5 As Double, z6 As
Double, z7 As Double, z8 As Double, z9 As Double
    Dim Zx As Double, Zy As Double, Zxx As Double, Zyy As Double, Zxy As Double
    Dim p As Double, q As Double

    z1 = myArray(dj(2) + 1, di(2) + 1)
    z2 = myArray(dj(1) + 1, di(1) + 1)
    z3 = myArray(dj(8) + 1, di(8) + 1)
    z4 = myArray(dj(7) + 1, di(7) + 1)
    z5 = myArray(dj(6) + 1, di(6) + 1)

```

```

z6 = myArray(dj(5) + 1, di(5) + 1)
z7 = myArray(dj(4) + 1, di(4) + 1)
z8 = myArray(dj(3) + 1, di(3) + 1)
z9 = myArray(dj(0) + 1, di(0) + 1)

Zx = (z2 - z6) / (2 * dX)
Zy = (z8 - z4) / (2 * dX)
Zxx = (z2 - (2 * z9) + z6) / (dX ^ 2)
Zyy = (z8 - (2 * z9) + z4) / (dX ^ 2)
Zxy = (-z7 + z1 + z5 - z3) / (4 * dX ^ 2)
p = Zx ^ 2 + Zy ^ 2
q = p + 1

Dim slope As Double

'Set slope in degrees
slope = Sqr(p)

'If slope is 0
If slope = 0 Then
    'Aspect is undefined
    getAspect = -1
ElseIf Zx = 0 Then
    'Aspect is north or south
    getAspect = (PI / 2) + (PI / 2) * (Zy / Abs(Zy))
ElseIf Zy = 0 Then
    'Aspect is East or west
    getAspect = PI + (PI / 2) * (Zx / Abs(Zx))
Else
    getAspect = PI - Atn(Zy / Zx) + (PI / 2) * (Zx / Abs(Zx))
End If
End Function

```

Matlabfunksjonen flowdirection

```

function [flowdir, dircount] = flowdirection(Z, tie_threshold)
%[flowdir, dircount] = flowdirection(Z, tie_threshold)
%Calculate flow direction (D8) for all non boundary cells in Z
%The directions are reported in powers of 2 starting from east:
% 1   2   4
% 128  0   8
% 64  32  16
% The value zero indicates a pit
% If there is a tie between two or more neighbours the sum of the
% directions is reported, thus a value of 255 indicates flat area.
% A threshold can be set so that angles >= maxangle*(1-tie_treshold)
% are treated as ties

if nargin < 2
    %default tie treshold, difference in h/l ratio less 0.1 percent
    tie_threshold = 0.001;
end
lbound = 1-tie_threshold;

%get number of rows and columns
[nrows, ncols] = size(Z);
%initiate the output flowdir matrix with NaNs
flowdir = NaN(nrows, ncols);

%initiate an array with distances to 8 neighbours
%that runs clockwise from northwest
dist = [sqrt(2), 1, sqrt(2), 1, sqrt(2), 1, sqrt(2), 1];

```

```

%Loop through all non-border cells and analyse nhoo
%(border cells will still be NaN/NODATA)
for i = 2:nrows-1
    for j = 2:ncols-1
        %extract focal cell
        c = Z(i,j);
        %extract neighbours to a one dimensional array
        %that runs clockwise from northwest
        nhoo = [Z(i-1,j-1:j+1) Z(i,j+1) Z(i+1,j+1:-1:j-1) Z(i,j-1)];
        %calculate the slope (delta_z/dist) between focal cell and
neighbours
        angles = (c-nhoo)./dist;
        %find the maximum positive angle
        maxangle = max(0, max(angles))*lbound;
        %find all neighbours with angle larger than maxangle
        hits = find(angles >= maxangle);
        %set the direction code
        flowdir(i,j) = sum(2.^(hits-1));
    end
end
if nargout > 1
    %if a name for the directioncount matrix is given
    %then create one
    dircount = NaN(nrows, ncols);
    %set all non border cells to zero
    dircount(2:end-1,2:end-1) = 0;
    %create a copy of the forwdir matrix
    tmpflowd = flowdir;
    %Now loop through the 8 directions (backwards) and increment the
    %dircount accumulator
    for i = 7:-1:0
        %calculate the direction code as 2 squared by i
        dircode = 2^i;
        %get index to all cells with tmpflowd>=dircode
        ix = tmpflowd>=dircode;
        %add 1 to the dircount matrix in the matching cells
        dircount(ix) = dircount(ix) + 1;
        %subtract dircode from the matching cells in tmpflowd
        tmpflowd(ix) = tmpflowd(ix) - dircode;
    end
end
end
end

```